# EYESEECAM Manual

September 19, 2007

# Contents

L	Fire	st Steps / Tutorial	7
1	<b>Asso</b> 1.1 1.2 1.3 1.4	embly and Operation         EyeSeeCam Parts         Start of operation         1.2.1         Binocular operation         Subject Preparation         Maintenance and adjustments	<b>9</b> 9 10 11 11 12
		1.4.1       Adjustment of the canoration laser         1.4.2       Replacement of lenses	12
2	Soft 2.1 2.2 2.3 2.4 2.5 2.6 2.7	wareRun the ProgramGraphical User InterfaceCalibrationDatabaseData OutputData Input2.6.1Profiles2.6.2Recorded VideostreamsOther Tools	<b>15</b> 15 17 18 18 19 19 19 20
3	<b>Drav</b> 3.1 3.2 3.3 3.4 3.5	vbacks of the Prototype         Operating System: Linux         Computer platform: MacBook         Eye Tracker         Head Camera (Optional)         The Software Application	<b>21</b> 21 21 21 21 22
II	Us	er Manual	23
4	<b>Calil</b> 4.1 4.2 4.3	bration         Mechanical Adjustment         Procedure Instructions         Quality Check	<b>25</b> 25 25 25

5       Recording       22         5.1       Data Files       22         5.1.1       MAT-file       22         5.1.2       TXT-file       22         5.1.3       Data Content       22         5.1.3       Data Content       22         5.2.1       AVT-file       22         5.2.2       PGM-file       33         5.2.3       DV-file       33         6       Mini Interface       33         7       Viewer       33         7.1       Data Browser       33         7.2       Exporting and Printing       33         7.3       Types of Data Views       33         7.4       Matlab Command Window       33         7.5       Data Browsing with Octave       33         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         9.1       Database       44         9.2.1       Patient       4         9.2.2       Examination       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials <th></th> <th>4.4</th> <th>Internals</th> <th></th> <th>• •</th> <th>•</th> <th></th> <th></th> <th></th> <th>26</th>		4.4	Internals											• •	•				26
5.1       Data Files       2         5.1.1       MAT-file       2         5.1.2       TXT-file       2         5.1.3       Data Content       22         5.2       Video Files       22         5.2.2       PGM-file       33         5.2.3       DV-file       34         5.2.3       DV-file       34         5.2.3       DV-file       35         6       Mini Interface       33         7       Viewer       33         7.1       Data Browser       33         7.2       Exporting and Printing       34         7.3       Types of Data Views       33         7.4       Matlab Command Window       33         7.5       Data Browsing with Octave       39         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2.1       Diffraction Pattern       4         9       Database       44         9.1       Database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.4       Loading Measurement       44 </th <th>5</th> <th>Rec</th> <th>ording</th> <th></th> <th>27</th>	5	Rec	ording																27
5.1.1       MAT-file       2         5.1.2       TXT-file       2         5.1.3       Data Content       22         5.1.3       Data Content       22         5.2       Video Files       22         5.2.1       AVI-file       22         5.2.2       PGM-file       33         5.2.3       DV-file       34         6       Mini Interface       33         7       Viewer       33         7.1       Data Browser       33         7.2       Exporting and Printing       34         7.3       Types of Data Views       33         7.4       Matlab Command Window       34         7.5       Data Browsing with Octave       37         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         9.1       Database Tab       4         9.2       Usage of the database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Res		5.1	Data Files												•				27
5.1.2       TXT-file       24         5.1.3       Data Content       22         5.2       Video Files       22         5.2.1       AVI-file       22         5.2.2       PGM-file       30         5.2.3       DV-file       30         6       Mini Interface       31         7       Viewer       32         7.1       Data Browser       32         7.2       Exporting and Printing       36         7.3       Types of Data Views       33         7.4       Matlab Command Window       36         7.5       Data Browsing with Octave       32         7.5       Data Browsing with Octave       33         8       Sequencial Control       44         8.1       Introductions to Profiles       48.2         8.2.1       Diffraction Pattern       44         9.2       Usage of the database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement       44         9			5.1.1 MAT-	file															27
5.1.3       Data Content       24         5.2       Video Files       22         5.2.1       AVI-file       23         5.2.2       PGM-file       33         5.2.3       DV-file       33         6       Mini Interface       33         7       Viewer       33         7.1       Data Browser       33         7.2       Exporting and Printing       34         7.3       Types of Data Views       33         7.4       Matlab Command Window       39         7.5       Data Browsing with Octave       33         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.1       Diffraction Pattern       4         9       Database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7			5.1.2 TXT-f	île															28
5.2       Video Files       22         5.2.1       AVI-file       23         5.2.2       PGM-file       33         5.2.3       DV-file       34         6       Mini Interface       33         7       Viewer       33         7.1       Data Browser       33         7.2       Exporting and Printing       36         7.3       Types of Data Views       33         7.4       Matlab Command Window       39         7.5       Data Browsing with Octave       36         8       Sequencial Control       44         8.1       Introductions to Profiles       48.1         8.2       Receivers       44         9.1       Database       44         9.2       Usage of the database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8			5.1.3 Data (	Content															28
5.2.1       AVI-file       22         5.2.2       PGM-file       33         5.2.3       DV-file       36         6       Mini Interface       37         7       Viewer       33         7.1       Data Browser       33         7.2       Exporting and Printing       36         7.3       Types of Data Views       33         7.4       Matlab Command Window       38         7.5       Data Browsing with Octave       39         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2.1       Diffraction Pattern       4         9       Database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       56         9.2.10		5.2	Video Files .																29
5.2.2       PGM-file       33         5.2.3       DV-file       36         6       Mini Interface       37         7       Viewer       32         7.1       Data Browser       33         7.2       Exporting and Printing       36         7.3       Types of Data Views       37         7.4       Matlab Command Window       38         7.5       Data Browsing with Octave       39         7.5       Data Browsing with Octave       39         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2.1       Diffraction Pattern       4         9.2       Usage of the database       4         9.1       Database       4         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results: Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44 <t< th=""><th></th><th></th><th>5.2.1 AVI-fi</th><th>ile</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>29</th></t<>			5.2.1 AVI-fi	ile															29
5.2.3       DV-file       30         6       Mini Interface       31         7       Viewer       32         7.1       Data Browser       32         7.2       Exporting and Printing       33         7.3       Types of Data Views       33         7.4       Matlab Command Window       33         7.5       Data Browsing with Octave       33         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2       Usage of the database       4         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.			5.2.2 PGM-	file															30
6       Mini Interface       3         7       Viewer       3         7.1       Data Browser       3         7.2       Exporting and Printing       3         7.3       Types of Data Views       3         7.4       Matlab Command Window       3         7.5       Data Browsing with Octave       39         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2       Usage of the database       4         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examin			5.2.3 DV-fil	e															30
7       Viewer       3         7.1       Data Browser       3         7.2       Exporting and Printing       30         7.3       Types of Data Views       33         7.4       Matlab Command Window       33         7.5       Data Browsing with Octave       39         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2.1       Patient       4         9.2.2       Examination       4         9.2.3       Measurement       4         9.2.4       Loading Measurement       4         9.2.5       Measurement esults : Trials       44         9.2.7       Plotting data       44         9.2.7       Plotting data       44         9.2.7       Plotting data       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9 </th <th>6</th> <th>Mini</th> <th>Interface</th> <th></th> <th></th> <th></th> <th>•••</th> <th></th> <th></th> <th>•••</th> <th></th> <th>•</th> <th></th> <th>•</th> <th></th> <th>•</th> <th></th> <th></th> <th>01</th>	6	Mini	Interface				•••			•••		•		•		•			01
7       Viewer       33         7.1       Data Browser       33         7.2       Exporting and Printing       34         7.3       Types of Data Views       33         7.4       Matlab Command Window       39         7.5       Data Browsing with Octave       33         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2       Usage of the database       4         9.2.1       Patient       4         9.2.2       Examination       4         9.2.3       Measurement       4         9.2.4       Loading Measurement       4         9.2.5       Measurement Results : Trials       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator       50         9.2.10       Examinator       50         9.2.9 </th <th>0</th> <th></th> <th>Interface</th> <th></th> <th>31</th>	0		Interface																31
7.1       Data Browser       3:         7.2       Exporting and Printing       3:         7.3       Types of Data Views       3:         7.4       Matlab Command Window       3:         7.5       Data Browsing with Octave       3:         8       Sequencial Control       4:         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4:         9.1       Database Tab       4         9.2       Usage of the database       4:         9.2.1       Patient       4:         9.2.2       Examination       4:         9.2.3       Measurement       4:         9.2.4       Loading Measurement       4:         9.2.5       Measurement Results : Trials       4:         9.2.6       File Names       4:         9.2.8       Data analysis       4:         9.2.9       System       5:         9.2.9       System       5:         9.2.10       Examinator       5:         9.2.10       Examinator       5: <td< th=""><th>7</th><th>View</th><th>ver</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>33</th></td<>	7	View	ver																33
7.2       Exporting and Printing       3         7.3       Types of Data Views       3         7.4       Matlab Command Window       3         7.5       Data Browsing with Octave       3         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2.1       Patient       4         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       49         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator       50         9.2.10       Examinator       50         9.2.10       Examinator       50         9.2.10       Examinator       50         9.2.10		7.1	Data Browser		• • • •		• •	• •	• •	• •		•	•••	•	• •	·	• •	•	35
7.3       Types of Data Views       33         7.4       Matlab Command Window       39         7.5       Data Browsing with Octave       39         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2.1       Patient       4         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator       50         9.2.10		7.2	Exporting and	Printing	• • • •	•••	• •	• •	• •	• •	• •	•	• •	•	• •	•	• •	•	36
7.4       Matlab Command Window       33         7.5       Data Browsing with Octave       33         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2.1       Patient       4         9.2.2       Examination       4         9.2.3       Measurement       4         9.2.4       Loading Measurement       4         9.2.5       Measurement Results : Trials       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator       50         9.2.10       <		7.3	Types of Data	Views	• • • •	•••	• •	• •	• •	• •	• •	•	• •	•	• •	•	• •	•	38
7.5       Data Browsing with Octave       39         8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2       Usage of the database       4         9.2.1       Patient       4         9.2.2       Examination       4         9.2.3       Measurement       4         9.2.4       Loading Measurement       4         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator       50         9.2.10 <td< th=""><th></th><th>7.4</th><th>Matlab Comm</th><th>and Window</th><th>• • • •</th><th></th><th>• •</th><th></th><th></th><th></th><th></th><th>•</th><th></th><th>•</th><th>•••</th><th>•</th><th>• •</th><th>•</th><th>39</th></td<>		7.4	Matlab Comm	and Window	• • • •		• •					•		•	•••	•	• •	•	39
8       Sequencial Control       4         8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2       Usage of the database       4         9.2.1       Patient       4         9.2.2       Examination       4         9.2.3       Measurement       4         9.2.4       Loading Measurement       4         9.2.5       Measurement Results : Trials       4         9.2.6       File Names       4         9.2.7       Plotting data       4         9.2.8       Data analysis       4         9.2.9       System       50         9.2.10       Examinator		7.5	Data Browsing	g with Octave	• • • •		•••	•••	••	• •	• •	•	•••	• •	• •	•		•	39
8.1       Introductions to Profiles       4         8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       4         9.2       Usage of the database       4         9.2.1       Patient       4         9.2.2       Examination       4         9.2.3       Measurement       4         9.2.4       Loading Measurement       4         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator       50         9.2.10       Examinator <th>8</th> <th>Seq</th> <th>uencial Cont</th> <th>rol</th> <th></th> <th>41</th>	8	Seq	uencial Cont	rol															41
8.2       Receivers       4         8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       44         9.2       Usage of the database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       56         9.2.10       Examinator		8.1	Introductions	to Profiles										•	• •				41
8.2.1       Diffraction Pattern       4         9       Database       4         9.1       Database Tab       44         9.2       Usage of the database       44         9.2       Usage of the database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       56         9.2.10       Examinator       56         9.2.10       Exami		8.2	Receivers																41
9 Database       43         9.1 Database Tab       44         9.2 Usage of the database       44         9.2.1 Patient       44         9.2.2 Examination       44         9.2.3 Measurement       44         9.2.4 Loading Measurement       44         9.2.5 Measurement Results : Trials       44         9.2.6 File Names       44         9.2.7 Plotting data       44         9.2.8 Data analysis       44         9.2.9 System       56         9.2.10 Examinator       56         A Troubleshooting       55			8.2.1 Diffra	ction Pattern	• • • •				• •			•		• •	• •	•			41
9.1       Database Tab       44         9.2       Usage of the database       43         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator       50 <b>III Reference Manual</b> 53 <b>A</b> Troubleshooting       54	9	Data	base																43
9.2       Usage of the database       44         9.2.1       Patient       44         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator       50         9.2.10       Examinator       50         A       Troubleshooting       53		9.1	Database Tab												•				44
9.2.1       Patient       43         9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       44         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       44         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator		9.2	Usage of the d	latabase															45
9.2.2       Examination       44         9.2.3       Measurement       44         9.2.4       Loading Measurement       47         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       44         9.2.7       Plotting data       49         9.2.8       Data analysis       44         9.2.9       System       50         9.2.10       Examinator			9.2.1 Patien	ı <b>t</b>															45
9.2.3       Measurement       44         9.2.4       Loading Measurement       47         9.2.5       Measurement Results : Trials       44         9.2.6       File Names       49         9.2.7       Plotting data       49         9.2.8       Data analysis       49         9.2.9       System       50         9.2.10       Examinator			9.2.2 Exami	ination															46
9.2.4       Loading Measurement       4'         9.2.5       Measurement Results : Trials       4'         9.2.6       File Names       4'         9.2.7       Plotting data       4'         9.2.8       Data analysis       4'         9.2.9       System       5'         9.2.10       Examinator       5'         A       Troubleshooting       5'			9.2.3 Measu	urement															46
9.2.5       Measurement Results : Trials       44         9.2.6       File Names       49         9.2.7       Plotting data       49         9.2.8       Data analysis       49         9.2.9       System       50         9.2.10       Examinator       <			9.2.4 Loadin	ng Measurement															47
9.2.6       File Names       49         9.2.7       Plotting data       49         9.2.8       Data analysis       49         9.2.9       System       50         9.2.10       Examinator       50         III       Reference Manual       53         A       Troubleshooting       54			9.2.5 Measu	urement Results :	Trials														48
9.2.7       Plotting data       49         9.2.8       Data analysis       49         9.2.9       System       50         9.2.10       Examinator       50     <			9.2.6 File N	lames															49
9.2.8       Data analysis       49         9.2.9       System       50         9.2.10       Examinator       50         III       Reference Manual       53         A       Troubleshooting       54			9.2.7 Plottir	ng data															49
9.2.9       System       50         9.2.10       Examinator       50         III       Reference Manual       53         A       Troubleshooting       54			9.2.8 Data a	analysis															49
9.2.10       Examinator       50         III       Reference Manual       53         A       Troubleshooting       54			929 System	m	••••		•••	•••	•••	•••	•••	•	•••	• •	•••	•	•••	•	50
III Reference Manual     53       A Troubleshooting     55			9.2.9 Syster	inator	• • • •		•••	•••	•••	• •	• •	•	•••	• •	•••	•	• •	•	50
III Reference Manual53A Troubleshooting55			9.2.10 Exam		•••		• •	•••	••	•••	•••	•	•••	• •	••	•	•••	•	50
A Troubleshooting 55	ш	Re	ierence Mar	านลไ															53
	Α	Trou	bleshooting																55
A.1 MacBook		A.1	MacBook .																55

	A.2	Cameras	55
В	Eye	Movements in 3D with Scleral Markers	57
	<b>B</b> .1	Marker Application	57
	<b>B.2</b>	Detection	58
С	Trac	king of Horizontal, Vertical, and Torsional Eye Movements	61
	<b>C</b> .1	METHODS	61
		C.1.1 Overview	61
		C.1.2 Pupil Detection	62
		C.1.3 3D Geometric Model Approach for Eyeball and Camera	62
		C.1.4 Geometric Model: Axially symmetric configuration	63
		C.1.5 Geometric Model: Rotation of the Coordinate System	64
		C 1.6 Geometric Model: A Calibrated Affine Transformation of the Image	65
		C 1.7 Measuring Torsional Eve Movements	66
			00
D	Use	r Environment	69
	<b>D</b> .1	The User "eyesee"	69
	D.2	Create a New Working Directory	69
	D.3	Create a New User	69
Ε	Con	figuration Files	71
	<b>E.1</b>	camera.xml	71
	E.2	iofile.xml	72
	E.3	ports.xml	72
	E.4	servo.xml	72
_			
F	Out	put Files	73
	F.1	Data in MAT-File	73
	F.2	Video in PGM-File	73
	F.3	Video in RAW-File	75
	<b>F.4</b>	Video in DV-File	76
	F.5	Video in AVI-File	76
G	Inni	it Files	77
G	G 1	Stimulation Profiles	77
	0.1	G 1 1 Basic Contents and Syntax	, , 77
		G12 Laser Pattern Activation	80
		C.1.2 Laser Fattern Activation	00 00
		C.1.4 Output of Voltage Signals for other Devices	00
		C.1.4 Output of voltage Signals for other Devices	01 01
		C.1.5 Recording an External Head Tracker	ð1
		G.1.6 Visual Stimulation	81
	<b>G</b> .2		82
Bi	bliog	raphy	83

# Part I First Steps / Tutorial

# **1** Assembly and Operation

# 1.1 EyeSeeCam Parts

The EYESEECAM video-oculography (VOG) system consists of

- a laptop
- the VOG mask
- an IEEE1394-Hub ("Firewire"-Hub)
- three firewire-cables
- and a transport case.

The Laptop is an Intel based Apple MacBook computer on which Linux is installed as the operating system. The main reason for choosing this hardware / software combination was the presence of an IEEE1394 connector with additional power supply from the laptop battery. There are some caveats of this combination that are listed in Chapter 3. IEEE1394 is a standard for transmitting (video) data that is used by EYESEECAM as its central technology. It is also called "FireWire" or "i.Link". The EYESEECAM VOG-system uses this standardized interface to transmit its video data to the laptop.



Figure 1.1: **VOG mask** (1) with VOG camera (2), infrared mirrors (3) and LEDs (4), and calibration laser (5).

#### The VOG mask consists of

- swimming goggles 1.1(1), used as base frame
- two infared-sensitive FireWire-cameras 1.1(2) with its particular firewire connectors 1.2(1) and its focusable lenses 1.3 (2)
- two infrared mirrors1.1(3), which reflect infared light to the cameras, but which are transparent for visible light
- infrared LEDs 1.1(4), which illuminate the eyes
- a calibration laser 1.1(5), which projects a pattern of light spots. These are used to calibrate the VOG-system.
- adjustment screws 1.2(2,3) which are used for positioning the cameras so that the bulbus appears in the middle of the video image.



Figure 1.2: **Camera connector and camera adjustment.** Plug the FireWire-cable into the firewire connector (1). The camera is adjusted vertically with adjusting screw (2) and horizontally with adjusting screw (3) so that the eyeball appears in the middle of the video image.

## 1.2 Start of operation

To start running the EYESEECAM VOG system for *monucular* use, plug one FireWire-cable into the camera that you want to use. Plug the other end of the cable into the laptop's firewire connector.



Figure 1.3: VOG camera (1). Plug the FireWire cable (3) into the camera's connector. When pluging or unplugging the cable, hold the camera (1) with your fingers on the white camera case. Whenever possible, avoid unplugging the FireWire connector to preserve the connection. Adjust focus of the eyeball's image with the adjustable lens (2). If the lens is rough-running, try to turn it a few times in the range ±180 to relax the silicone rubber spring.

Attention! When plugging the cable into the camera's connector, hold tight the camera at its white casing (1.3 (1)) to avoid mechanical damage of the adjustment mechanism (1.3 (3)). Whenever possible, avoid unplugging the FireWire connector to avoid abrasion of the connector.

#### 1.2.1 Binocular operation

For *binocular* operation, both cameras have to be connected to the laptop.

Attention! Due to malfunction of the provided FireWire hub (Belkin F5U526), a certain sequence has to be maintained, when connecting two cameras to the laptop. It does not matter, which of the hub's connectors you use.

- 1. First, connect only one camera to the hub.
- 2. Connect the hub with the laptop.
- 3. Wait for approximately five seconds
- 4. Connect the second camera to the hub

### **1.3 Subject Preparation**

Use the adjustment clip to adapt the elastic tape so that the mask is mounted tightly but comfortably to the subject's head. Attach the VOG mask to the subject's head). After starting the EYESEECAM software as described in Chapter 2, use the adjusting screws (1.2 (2,3)) to center the eyeballs in the video images shown in the video monitors of EYESEECAM. Turn the focusable lenses (1.3 (3)) so that you see sharp images of the eyes in the video monitors. If the lenses are rough-running, try to turn them a few times in the range  $\pm 180$  to relax the silicone rubber spring.

Eyes do not only move in the horizontal and vertical directions (2D) but they also move in the torsional direction around the line of sight. If all three components are taken into account the eye movements are measured in 3D. Pupil tracking alone gives a 2D measurement. The third torsional component can be determined from either natural or artificial landmarks that are visible on the sclera or on the iris. Currently, EyeSeeCam requires dark artificial landmarks that need to be applied to the sclera with a cosmetic pigment and a sterile surgical pen. This procedure is described in detail in Chapter B.

After all these preparatory steps the calibration procedure can be started as described in Chapter 4, and subsequently, the examination can be started either with the help of the integrated database (as described in Chapter 9) or with the methods described in Chapter 2.

#### 1.4 Maintenance and adjustments

#### 1.4.1 Adjustment of the calibration laser

To adjust the calibration laser, switch it on as described in Chapter 4 and turn the laser's horizontal and vertical adjustment screws (1.4 (2,3)) so that the middle point of the laser pattern is in straight-ahead position.



Figure 1.4: **Calibration laser (1).** To adjust the laser to be aligned with gaze straight ahead, turn the horizontal (2) and the vertical (3) adjusting screw with a small screw driver.

#### 1.4.2 Replacement of lenses

If you want to measure with the full frame rate of 500Hz, the vertical range of the video image is limited. Depending on your measuring setup, the vertical eye positions may run out of range. To

avoid this limitation, you can use lenses with shorter focal length. We recommend lenses with 8mm focal length for measuring with 500Hz. To replace the lenses, unscrew the original lens from the lens holder. Remove the silicone rubber spring (1.3) from the original lens and put it onto the new lens. Now screw the new lens on the lens holder.



Figure 1.5: infrared mirror (1) and mirror carrier (2). To preserve the infrared mirror when transporting the system, release the mirror carrier by loosening the retaining screw (3). Warning! DO NOT try to adjust the mirror by turning the mirror carrier! It's only a mounting element!

# 2 Software

#### 2.1 Run the Program

When the computer starts up the user "eyesee" is logged in automatically, see figure 2.1.

- 1. Make sure the hardware is pluggerd in.
- 2. Double-click on the *desktop* icon "EyeSeeCam".
- **Troubleshooting** Computer hangs during boot sequence. If it states "... has not been checked for 60 days. Check forced.", wait until this is done. Otherwise, if it really hangs, see section A.1.

**Troubleshooting** If no cameras were found, see figure 2.2 and section A.2.

#### 2.2 Graphical User Interface

The Graphical user interface (GUI) is tab based. The main tab is *Eye Tracker*. This tab contains other tabs for single cameras (Left, Right, ...) as well as tabs for combinations of cameras (Binocular, ...). The terms "left" and "right" always refer to the person who wears the eye tracking goggles.

You can get some help on many widgets by moving the mouse pointer over it. After a short delay a tool tip appears with some explaining text about the function of that widget. In case there is a keyboard shortcut for e.g. a pushbutton, that is mentioned in the corresponding tool tip. There are also keyboard shortcuts for the menu items.

The status bar at the button of the main window usually shows a message in response to a user action. These messages are logged on tab "Log" in the lower right area of tab "EyeTracker".

If you have connected the eye tracking camera of the left eye, the tab "Left" (within tab "Eye Tracker") is enabled. This tab is divided in to three areas, each of them again is a row of tabs. The upper left row of tabs ("Processing", "Raw", "Status") shows the video of the corresponding camera plus some information about its dimensions, etc. The raw video, i.e. like it comes from the camera, is displayed on tab "Raw", while you can watch the intermediate results of the processing algorithms in tab "Processing".

The widget which shows the video has some hidden but important features. Let's call it *videobox*. On right mouse click into this videobox a context menu pops up. Here you can select what you want to see. Un-check pupil center, and the green cross at the pupil center disappears.



Figure 2.1: The desktop after start-up. There are desktop icons "EyeSeeCam" and "EyeSeeCam Updater".



Figure 2.2: An empty window appears if no cameras were found.

Check globe, and watch the internal model of the eyeball. While the videobox ignores *single* left-mouse-button clicks it puts the videobox to full-screen on double-click. Another double-click returns to the normal view. When you press Ctrl+Alt (Ctrl+Apple on MacBooks) you can drag a rectangle in the videobox with the left mouse button. The rectangle is applied when you release the mouse button, its interpretation depends on its size and position:

- A big rectangle that extends from the left half of the image to the right half is interpreted as a region of interest (ROI). That ROI can also be set on tab "ROI" using the four sliders. The ROI should contain the pupil entirely for all possible gaze directions, additionally we recommend to include also iris and some white sclera. Usually it should not be necessary to change the ROI, except when dark shadows appear at the image border or when the or if make-up disturbs the pupil detection.
- A smaller rectangle within the left or right half image (it does not cross the vertical midline) defines the search area for a *sclera marker*. Markers are artificial black-pigment dots applied to the white sclera. Marker detection is used to determine the torsion (roll) of the eye around the axis of gaze. EyeSeeCam is capable to detect two markers per eye: one on the left side of the iris, the other on the right side. Before you drag the rectangles that define the marker search areas, (1) make sure the system is calibrated and (2) advise the subject to do a straight ahead fixation. Then drag the rectangles that contain the entire maker plus a bit of surrounding white sclera. It is essential to the maker detection algorithm that the border of the rectangle is drawn over background.
- A zero-size rectangle (press-and-release) clears the search area of the marker of the corresponding side. No marker detection is performed then.

To the right of this row of tabs there is another row of tabs ("ROI", "Thresholds",  $\ldots$ ) where you can control the parameters of eye tracking.

Below these two rows of tabs, in the center row of the window, you find the plots ("Position",  $\dots$ ).

**Troubleshooting** If once EyeSeeCam shows some instability at program start or when you just click through the GUI, especially after an update, then delete or rename file .eyeseecam/appearance.xml in your home folder. At next start, EyeSeeCam appears with standard settings.

### 2.3 Calibration

On tab "Eye Tracker" there is a sub-tab "Calibration" in the lower left area. Do not confuse it with the tab "Servo Calibration" (optional, only with servo driven head mounted camera). "Calibration" defines the relation of the pupil position in the video image and gaze direction (horizontal and vertical eye position) while "Servo Calibration" defines the relation of the pupil position and the servo positions to pan and tilt the head mounted camera.

Press "Start" in tab "Calibration". The laser, located on the goggle between the eyes, is lit and projects a target pattern of red laser points to the wall in front of the subject. You may press "Abort" (Key Esc) any time to break the procedure and discard the acquired data. A progress bar shows the progress. During the calibration procedure the subject should fixate five points: the center-point of the diffraction pattern and the four nearest neighbors (left, top, right, bottom). The Order does not matter.

There are two tested strategies:

- 1. The subject performs a short fixation and "jumps" to the next target. When all five target points have been fixated this is repeated until the laser points disappear.
- 2. The subject performs five long fixations. The operator advises the next target at 20%, 40%, 60%, and 80% progress.

The calibration may succeed or fail. In case of success

- the bullet of the radio-button group "Parameters" is set to "Calibrated",
- in the videobox
  - the target pattern (the group of little red crosses) has moved to a new position and
  - a pink figure appears, connecting the five (really all five?!) central little red crosses of the target pattern.

Switch on the laser target pattern again (the check-box below the progress bar) to check the calibration. Advise the subject to fixate the target dots again, this time all laser dots that are visible. Double-click into the videobox to see the details on full-screen. The green cross indicating the pupil center must jump to the red cross positions indicating where the pupil should be in that gaze position according to the calibrated internal 3D model.

In case the calibration failed, check (1) whether the cameras are still operating, (2) the mechanical adjustments, (3) no shadows inside the ROI, (4) no reflections on the cornea big enough to irritate the pupil detection.

See chapter 4 on page 25 for details.

#### 2.4 Database

The program provides the opportunity to create and carry out a database. This database helps you to arrange the examination of your patients and the analysis or evaluation of your measured data. You can enter the patient's data and you can choose your kind of examination or measurement. To evaluate the measured data several specialized analysis scripts are available.

See chapter 9 on page 43 for details.

#### 2.5 Data Output

Output files are controlled in tab "Record".

If you check a camera in button-group "Videostream", the videostream of this camera is written to file when recording. IMPORTANT: You may create very lage files in quite a short amount of recording time! Check for enough disk space. The current version creates a <basename>.pgm file. The format is multi-image Portable Grey Map. It contains the uncompressed video images. Most graphical tools read PGM format files, but unfortunately many of them stop reading after the first frame. Actually the multi-image PGM is just a simple concatenation of single-image PGM. See also section F.2 on page 73.

With standard setting, a <basename>.mat file is created that contains the results. The format is compatible to MATLAB version 4. It can be imported by MATLAB and Octave.

A <basename>.txt file may be created that contains the same data as the <basename>.mat. It is a tab-separated ASCII table. It can be imported by Excel, Gnumeric, gnuplot, and other spread-sheet programs. The '#' character is used to mark comments as this works fine for gnuplot.

#### 2.6 Data Input

#### 2.6.1 Profiles

Many applications of VOG demand the measurement of eye movements combined with a given stimulus. Such a stimulus may be a voltage output as a function of time.

EyeSeeCam offers an interface to control some program parts by a profile. A profile is a file in MATLAB 4 format that contains mainly a time series. This time series is a matrix that has a column "Time", and other columns with the corresponding values, e.g. volts. Section ?? describes how to create the profiles.

Load an existing profile with button "Load" on tab EyeTracker/Paradigm. Make sure you have the required euipment, e.g. USB-DUX, connected and ready to use.

Working with the database, the profile that corresponds to the examination and trial is automatically loaded.

When the user starts the profile, at the same time *recording* begins automatically (see tab EyeTracker/Record). Recording is terminated when the profile is finished (100%) or when the user aborts the profile.

#### 2.6.2 Recorded Videostreams

Recorded videostreams can be re-played. Drag and drop the video file on the EyeSeeCam desktop icon. Make sure the camera system is not connected to the computer.

The file format is automatically detected if it is supported. All file formats available for recording are supported also for the player function. We recommend \*.pgm for recording the videostreams of eye tracking cameras, and \*.avi other cameras.

Controls for the player are in tab EyeTracker/Play (bottom left area). Buttons "Start" and "Stop" continue and halt the videostreams. If the videostreams are halted, the user may step one frame forward and back with the buttons "Next" and "Back". The latter buttons have an auto-repeat mode, so keep pressing the button down for single-step mode. The button "Reload" just queries the current frame(s) again, which is usefull if you have changed an image processing parameter.

In case the file contains more than one videostream, the buttons above effect *all* videostreams, whereas the controls below effect only the selected one. The slider "Position" can be used to

go to a desired position. The current program version requires a pre-scan of the file in order to make this feature avaliable. So press "Start" first, then after the file has been played once, press "Stop" and drag the slider.

To analyse the pupil positions in a videostream, we recommend the following procedure: (1) Use "Start" and "Stop" to go the the file position where the analysis should start. (2) Press "Record" in tab "Record", then press "Start" in tab "Play". (3) Press "Stop" in tab "Play", and "Stop" in tab "Record".

## 2.7 Other Tools

??EyeSeeCam Updater

# 3 Drawbacks of the Prototype

## 3.1 Operating System: Linux

- EyeSeeCam is a multi-treading application. In order to assign different priorities to the various threads, Linux requires the program to be started with super-user permissions. We have configured the system that way. Please use the provided desktop symbols to start EyeSeeCam. Without super-user permissions all threads run in normal priority, causing poor performance and severe disturbance on usability.
- EyeSeeCam needs your computer to run with full perfomance. Make sure that CPU frequency policy is set to "Performance", not "Dynamic".
- When plugging in the TV adapter (Terratec Cinergy Hybrid T) into USB, there is a delay of about half a minute! This is due to the corresponding USB driver modules are loaded automatically when detecting the hardware. Do not remove and re-plug the TV-USB-adapter, since re-loading of the driver modules does not work and may cause system failure.

## 3.2 Computer platform: MacBook

Why a MacBook? We searched for a mobile, light-weighted computer platform, that has at least one full-featured IEEE1394 plug. Unfortunately, most laptop computers are equipped only with the smaller 4-wire plug, while we require the bigger 6-wire version, since the additional two wires supply the power for our system. Laptops of another vendor failed with respect to the current limitation.

The ability of MacBooks to host the Linux operating system is quite new. Hopefully there will be updates available to get rid of the most annoying drawbacks of the Linux/MacBook combination.

## 3.3 Eye Tracker

??design, ??hot mirrors, sun and spotlights, ??slippage

# 3.4 Head Camera (Optional)

??limitations of recording, compressed DV vs. uncompressed format

??system latency

### 3.5 The Software Application

The software is still under developement. There are some issues where you might find the program not do what you expect.

- When a file (\*.pgm, \*.dv, \*.avi, \*.etd) is given as an argument to EyeSeeCam, it is analysed for video streams. For very large files this may take some time. The splash screen states "Searching for cameras..." meanwhile.
- The Graphical User Interface (GUI) is under development progress from a user interface for developers to a simple-to-use application: You may want more convienient features where there is just manual adjustment of poorly documented parameters.
- Configuration files are read at program start. No function to write back a modified configuration is implemented yet.

# Part II User Manual

# 4 Calibration

### 4.1 Mechanical Adjustment

- **Best fit of the goggles** User and operator must be aware of the slippage problem from the beginning. The goggles are placed on the skin, and the skin is not at all fixed onto the head. The goggles might slip slowly on the skin. The found the goggles that we use as basis for our system to minimize both undesired effects. We recommend to advice the user: take some time to find the best fit of the goggles. Check for even contact to the face. People with small noses may have a gap over the bridge or root of the nose, others should not. Check for symmetry and level. If the user reports a local discomfort, it may be better to strengthen the belt instead of loosening it.
- **Focus** The lense of each camera has a screw thread to allow manual focus. There is a transparent piece of hose that fixates the lense by means of friction and elasticity. As the distance of the camera/mirror to the user's eye is varies from person to person, the focus should be checked and usually needs adjustment. For this purpose double-click into the videobox that shows the corresponding eye in order to view it on full screen. Turn the plastic ring that holds the lens and turn it. Use enough force to overcome the friction of the hose. Optimize for a sharp edge of the pupil. Visible structures on the iris might be helpful. Do not focus on other objects like eyelashes!
- **Pan and tilt** Use the knurled screws on the back plane of the camera(s) to adjust pan and tilt. If you watch the corresponding eye in the videobox, you can see it move horizontally or vertically when you turn the screws. Adjust the camera in order to have the eye in the image center.<sup>1</sup>

# 4.2 Procedure Instructions

??

## 4.3 Quality Check

??

<sup>&</sup>lt;sup>1</sup>Actually it is the eyeball center that should be in the image center. You can test this if the user looks straight into the camera. The camera is only visible under high contrast conditions, i.e. there is a dark background straight ahead and the camera is in spot light. Then the mirror images of the cameras appear in about 100 mm distance in front of the user. Fortunately, we have a calibration procedure, so we do not have to care about such fine adjustments—except for some reason the calibration is not possible!

#### 4.4 Internals

**Assumptions** The angular distance of the top, left, bottom and right laser target point to the center target point given, default value is  $8.5^{\circ}$ . The orientation is adjusted, i.e. top-center-bottom laser target points are really vertical (in upright head position).

The distance of the system/eyes to the projection wall or screen is unknown. Instead, far distance is assumed. Parallax is ignored.

**Acquisition** During the calibration procedure a given number of valid pupil coordinates is acquired. So if the cameras run with more frames per second the procedure takes less time. If the user blinks or for any reason there is no valud pupil detected, the procedure takes more time.

A click on the button Start in tab Calibration starts the calibration procedure for *all* eye tracking cameras. The progress bar shows the progress of *one* of these cameras, usually camera "Left" if available. Thus, it is possible to confuse the program by closing an eye lid.

In normal operation the user may have selected automatic or manual slippage correction, but calibration is always based on the original data (ignoring slippage correction).

**Cluster** The array of pupil positions is passed to a 2D cluster algorithm. It tries to group the pupil position into five clusters that meet criteria about distance, extent and number of samples. In case of failure the algorithm returns without result.

The five groups are identified as the upper, leftmost, lower and rightmost cluster, and the remaining cluster is identified as the center. The final result of the cluster algorithm is the horizontal and vertical median position of each cluster.

**Parameter Adjustment** The five pupil positions that result from the cluster algorithm are compared with five corresponding pupil positions that are reverse-calculated using the built-in geometric model. The comparison considers

- angular orientation  $\rightarrow$  angle of rotation,
- average distance  $\rightarrow$  scale factor,
- location of the center-of-mass  $\rightarrow$  horizontal and vertical shift.

A 2x3-matrix is built up on these four parameters to transform the un-calibrated pupil positions to calibrated pupil positions. This affine transformation is not capable of shear, rotation larger than about 45°, anisoptropy, or non-linearity.

For further details see chapter C on page 61.

# **5 Recording**

The tab "Record" is located in the lower right area of tab "Eye Tracker". You can record

- videostreams of one or more cameras to a single file, as well as
- *data* of pupil and marker coordinates, state calibration laser, servo positions and many more to a data file.

Click on button "Record" to start recording. A timer (above the "Record" button) counts up the seconds. Click on "Stop" to stop recording.

					Options for Recordin 🗵
					_Videostream
Record Annotati	ons Log				Format .avi 💌
	Basename –				Data logge ngm
F Left	20070818-0	)0		New	MAT-fi .avi
F Right					
<b>₩</b> 2		0:00	0:00		
	Options	Record	Stop	Plot	Hide

Figure 5.1: Tab "Record". To record videostreams from several cameras, you have to check each camera. Click "Browse" to change the <br/>basename> of the recorded files. Click "Options" to select video and data file formats.

## 5.1 Data Files

There are two formats of data files: a MAT-file and a tab-separated ASCII table. The <base-name>.mat file is created by default, whereas the <basename>.txt file is not.

#### 5.1.1 MAT-file

The MAT-file corresponds to MATLAB Version 4 file format. It can be opened with MATLAB and Octave.

The MAT-file contains a sequence of matrices. Each matrix starts with a fixed-length 20 byte header that contains information describing certain attributes of the matrix, e.g. number of rows and columns.

Start MATLAB and open <basename>.mat.

The main matrix is named Data. It uses the numeric format IEEE Little Endian with single precission (32 bit) floating point numbers. Corresponding to the columns in "Data" there are text matrices DataNames and DataUnits added that contain a name and a unit for each column. For example, the first string in DataNames is "Time", and the first string in DataUnits is "s".

Other text matrices are added for convienience. Their name begins with "Eval", e.g. EvalDataColumns. Im MATLAB you can type

>> eval (EvalDataColumns)

to evaluate the string. In this case a structure col is created. Its components, e.g. col.Time, equal the column count. col.Time is set to 1, as the column time is the first column of the Data matrix. We strongly recommend to use the literal names in scripts rather than the numeric column index, because the number of columns may vary from MAT-file to MAT-file.

#### 5.1.2 TXT-file

The TXT-file contains the same data as the MAT-file, except the Eval\* scripts. It is a tabseparated ASCII table. The lines of the header begin with the '#' which is the comment sign of Gnuplot. The text of MAT-file's DataNames and DataUnits is printed in the rows right above the data rows. An additional row above just holds numeric column indices (1, 2, 3, ...)for convienience.

#### 5.1.3 Data Content

**Time** in units of "s" (seconds). The elapsed time since button "Record" was pressed. It equals one of the values LeftTime, RightTime, Left2Time, or Right2Time.

**Left\***, **Right\***, **Left2\***, **Right2\*** Columns with these prefixes correspond to a given eye tracking camera. "Left" refers to the camera at the subject's left eye. In case there is a second camera at the left eye that has the prefix "Left2".

**LeftTime, ...** in units of "s" (seconds). The elapsed time of the current frame of the given camera since button "Record" was pressed. The underlying timestamp is set about 0.05 ms before the frame is available for image processing.

**LeftEyeHor, LeftEyeVer, ...** in units of "deg" (degree). Horizontal and vertical eye position. Actually these are the component of a rotation vector that transforms reference gaze direction to current gaze direction. The rotation vector directs parallel to the rotation's axis, and the rotation vector's absolute value is the rotation angle in degree. The third (torsional) component of this vector is zero.

**LeftEyeTor, ...** in units of "deg" (degree). Torsion of the eye around the axis of gaze. Actually this is the only non-zero component of a rotation vector that rolls the eye based on the

sclera marker detection, before another rotation vector (see LeftEyeHor, LeftEyeVer, ...) rotates the gaze direction. To get the rotation vector that does the complete transformation at once, you must combine these two rotation vectors in correct order.<sup>1</sup>

**LeftPupilCol, LeftPupilRow, ...** in units of "pixels" (pixel width and pixel height, resp.). Image coordinates of the pupil center, based on the center-of-mass algorithm. The coordinate system's origin (zero) is located in the image center. Positive values are to the left (subject's view) and up.

**LeftPupilEllipseCol, LeftPupilEllipseRow,...** in units of "pixels" (pixel width and pixel height, resp.). Image coordinates of the pupil center, based on an ellipse-fit algorithm<sup>2</sup>. The coordinate system's origin (zero) is located in the image center. Positive values are to the left (subject's view) and up.

#### 5.2 Video Files

It is possible to record video of all cameras into serveral file formats. The default format is AVI, but it is possible to select RAW, PGM or DV. You can record as many cameras as you like into one single file. EyeSeeCam is able to load all these formats and display the stored cameras as virtual live cameras.

#### 5.2.1 AVI-file

AVI is the default container format to store videos. Each camera is stored in an extra video track within the AVI container. Most Video-Players support only one video track and therefore display only the first track. We recommend to use the VLC media player<sup>3</sup> to display recorded AVI files with all video tracks. Windows users are encouraged to download the latest DivX<sup>4</sup> codec to view the videos in Windows Media Player and Microsoft Powerpoint. Each videotrack is compressed by default with a FFMPEG MPEG-4 (DivX) codec with maximal 8 MBit/s data rate. It is not yet possible to change the compression settings. You may prefer this format to capture compressed and easy-to-use videos with the head mounted camera or for presentations.

<sup>&</sup>lt;sup>1</sup>A "Eval\*" script for MATLAB will be added in future versions.

<sup>&</sup>lt;sup>2</sup>In case no ellipse-fit algorithmellipse-fit algorithm is used, the current software version puts the values LeftPupil-Col, LeftPupilRow, ... here again.

<sup>&</sup>lt;sup>3</sup>http://www.videolan.org

<sup>&</sup>lt;sup>4</sup>http://www.divx.com



Figure 5.2: The VLC media player displays all recorded video tracks of one single AVI file. VLC is available for various operating systems (here: Mac OS X).

#### 5.2.2 PGM-file

The "Portabe GrayMap" (PGM) format is attended to store every frame and additional meta information directly to disk. It provides no compression or color conversion, but delivers the data in exactly the same way as it was grabbed form the camera. You may prefer this format, if you like to examine the videostream by your own and can't accept any distortion, caused by video compression. The Grayscaled VOG eye cameras are stored as 8bit graymap, analog color cameras in YUV422 format are stored as 16bit graymap.

#### 5.2.3 DV-file

The "Digital Video" (DV) format is only selectable, if you use an analog video to digital video (AV/DV) converter, like Canopus ADVC-55. This converter compresses the analog video by default with no additional CPU load. The compressed images from each camera are stored interleaved into a single \*.DV file.

# 6 Mini Interface

The default EyeSeeCam user interface could be to cluttered and difficult, if you want to create simple recordings with your head mounted cameras. The EyeSeeCam.mini provides an easy-to-use interface for the main tasks you have to do in a mobile situation:

- View all camera, to check correct capturing of the eyes, sharpness and image formation.
- Control servo actuators and start calibration procedure.
- Start and stop recordings.

The mini interface was designed for simple navigation. It is structured as a simple tree menu, similar to the one you find in your Apple iPod music player. The root menu presents you a list of five items. As you will see, the items are either options to enable or disable something, actions or sub menus. Sub menus are marked by an arrow on the right side. The last item in each sub menu leads you back to the upper menu. When you leave the root menu, you see the last viewed camera. It is possible to use your mouse or the up and down arrow keys to navigate. Active items are marked in orange. Press the left mouse button, the return key or the right arrow key to enter. Press the right mouse button or the left arrow key to leave a sub menu.



Figure 6.1: You can start the mini interface by selecting "Run EyeSeeCam.mini" in menu "Settings". The mini interface appears in a new window. To return to the default interface, select "Classic interface" with a mouse click or using the arrow keys.

Activate the "View Camera" sub menu item in the root menu and press the left mouse button, the return key or the right arrow key. You can choose between all attached cameras to view a camera. While viewing a single camera it is possible to use the mouse wheel to switch between all cameras.

View Camera		View Camera A
Control & Record		View Camera B
Settings		View Camera C
Classic Interface		◄ Back
<ul> <li>Back</li> </ul>		

Figure 6.2: "View Cameras" sub menu.

The "Control & Record" sub menu lets you start and stop the servo actuators, start and stop record, start the calibration procedure and switch the laser pointer. The mini interface always uses the custom calibration. The calibration is automatically stored and loaded on next start. When pressing "Start Record" the mini interface always creates a new AVI file with MPEG4 compression. The VOG data is stored in a MAT file. The filename has the format "yyyymmdd-hhmmss".

The "Settings" sub menu lets choose, which camera should be recorded (default "C" & "D"), if any filter should be applied to servo movement and if the mini interface should be displayed on EyeSeeCam startup by default. Every time you click "Save appearance" in the classic interface, EyeSeeCam will start with the classic interface at the following progam launches.

Start Servos	Record Camera 📃 🕨
Start Record	Servo Filter 🔹 🕨
Calibrate Servos	This interface on startup
Laser Pointer	Back
Back	

Figure 6.3: "Control & Record" sub menu lets you start and stop the servo actuators, start and stop record, start the calibration procedure and switch the laser pointer. The "Settings" sub menu lets choose, which camera should be recorded

# 7 Viewer

The EYESEECAM program records eye movement data to Matlab (mat) files. EYESEECAM is shipped with an external stand-alone Matlab program named "ezplot" that enables the user to view the contents of such a file as well as to browse through the saved data by using the "Zoom" and the "Hand" functions from the Matlab figure toolbar. With this tool the data can also be printed or exported to eps, pdf or other image files. Immediately after a data file is recorded it can be viewed by pushing the "Plot ..." button that is located in the "Record" tab in the lower right corner of the graphical interface (see Figure 7.1).



Figure 7.1: The data Viewer can be started from the graphical interface by pressing the "Plot ..." button that is located in the lower right corner of the "Record" tab.

The "Plot ..." button activates a file browser dialog in which the desired Matlab file can be selected (see Figure 7.2).

🗙 Open File		? 🗆 🗙
	) 😭 🔌 📔 /home/eyesee/eyeseecam/	<u>·</u>
🔂 public_html	20070411-00.mat	
🤪 Desktop S	20070415-00.mat	
I Desktop	eveball affine matrix.mat	
Documents	servocalib.save.mat	
合 Home Folder		
Storage M	Location: 20070415-02.mat	<u>⊠o</u> pen
💓 Network F	Eilter: MAT-files (*.mat)	💢 <u>C</u> ancel

Figure 7.2: The "Plot ..." pushbutton opens a file browser dialog in which a Matlab file can be selected that will opened by the external data viewer.

After a file is selected and the "Open" button is pressed the external viewer (see Figure 7.3) is started<sup>1</sup> and the data contents are displayed on a new graphical user interface that mainly consists of three plots, one for the horizontal, one for the vertical, and one for the torsional eye movement component.



Figure 7.3: The external data viewer displays horizontal (top), vertical (middle), and torsional (bottom) eye movement components. By default, it plots the uncalibrated eye position in pixel units over time.

In each of the subplots the eye position data are plotted over time in uncalibrated pixel units. The "Zoom" function that is located in the toolbar can be used to view the data contents at every level of detail required by the specific analysis. With the "Hand" tool the data can be dragged in either direction while the time axes of the subplots remain synchronized. At every viewing step, the current display contents can be printed or exported to a high-quality image file format like

<sup>&</sup>lt;sup>1</sup>The external data viewer is a stand-alone program that can also be started from a shell by entering a command like: ezplot mydata.mat

pdf. This functionality is standard for Matlab Figures, and therefore, it is not documented in great detail here. It is also standard Matlab functionality to export the thus plotted data to pdf, eps and a number of other formats. Please refer to the Matlab documentation for further details on the plotting and graphical capabilities.

Part of the Matlab functionality like, e.g., the export functions, can be accessed from the standard "File" menu of the viewer application. The functionality that is specific to this data viewer can be accessed from the "Actions" menu (see Figure 7.4).



Figure 7.4: The data viewer functionality can be accessed from the "Actions" application menu. From here, the control can be transfered to a Matlab command window with "Command ...", and also, different data viewing types can be activated with the menu "Plot >". With "File Open ..." a new Matlab data file can be opened for display (In the current version this function doesn't work properly due to a Matlab bug). The menu "Show PDF >" provides access to pdf export and viewing capabilities. With "Exit" the ezplot Matlab data viewer application can be terminated. The "File" menu is common to all Matlab applications. This menu was not altered and its use is documented in Matlab.

#### 7.1 Data Browser

The external stand-alone Matlab application ezplot was designed to support the EYESEECAM user with data browsing and monitoring capabilities. Immediately after a trial data set is recorded in the main EYESEECAM application it can be viewed for control purposes by pressing the "Plot ..." pushbutton (see Figure 7.1). When the "(+) Zoom" function is pressed in the toolbar a data region can be selected with the mouse in order to zoom in to the selected region. The time axes of all other subplots remain synchronized. In Figure 7.5, for example, one of the saccades from Figure 7.3 is displayed in a zoomed-in version. After this step the "Hand" tool can be used from the toolbar in order to scroll through the data, e.g., in the left and right

directions. Again, all other subplots remain synchronized. At each step, the user can return to the original view as displayed in Figure 7.3 by klicking with the right mouse button in any of the three subplots. Just like in Matlab, a pup-up menu appears in which "Reset to original view" can be selected.



Figure 7.5: Browsing through the data with the Matlab zoom and scroll functions. The zoom function (left) can be activated by pressing the magnifying glass with the (+) symbol on the toolbar. After this step a region can be selected in one of the subplots. In this example one of the saccades from Figure 7.3 was selected. When the hand symbol is pressed in the toolbar (right) the data can be scrolled from left to right by klicking in one of the subplots and dragging the mouse in the horizontal direction. The time axes of the individual subplots remain synchronized both during zooming (left) and scrolling (right). Plese note that the toolbar symbols for the magnifying glass and the hand are depressed in the left and right figure parts, respectively.

## 7.2 Exporting and Printing

Command	
Plot	Þ
File Open	
Show PDF	Þ
Exit	

With ezplot it is possible not only to display and browse EYESEECAM eye movement data on the screen but also to print the displayed data plots on a printer. Related to this printing functionality is an export functionality, that serves to generate image files in a number of standard formats like pdf and eps. This is a convenient means to exchange high-quality data plots with other programs like Word, PowerPoint or LATEX. pdf and especially eps files are highly welcome by publishers as the standard file format for scientific data plots. eps files can also be imported into graphi-

cal programs like Adobe Illustrator which in turn can serve as an intermediate step for integrating a figure into Microsoft Word or PowerPoint.
humbrialis	2007-04-15 00:27:31 273109
	0 5 93 Time (s) 20 20

Figure 7.6: The data plots from Figure 7.3 can be exported to a pdf file. Such a pdf file can be viewed with a program like kpdf or Acrobat Reader (acroread).

When the user presses the "Show PDF >" button the current contents of the display are exported to a pdf file. This file is located in the same directory and –apart from the additional suffix .pdf- it is also given the same file name as the original Matlab data file. This pdf file can then be viewed and subsequently printed either with Acrobat Reader (acroread) or with kpdf which is also a pdf viewer (see Figure 7.6).

The standard Matlab "File" menu provides additional printing and exporting possibilities. With the menu "Save as ..." it is possible to export the current display contents to a number of file formats that are supported by Matlab. A file browser is opened in which the desired file name can be entered. The desired image file format can be chosen from the drop-down box entitled "Save as type:".

		X Save As
		Filter
		Ĭ
		Directories Files
		Phone/degeoc/egeocom/. A [ ] A //home/degeoc/egeocom/.
New	Þ	
Open	<ctrl>-0</ctrl>	
<u>C</u> lose	<ctrl>-⊎</ctrl>	Save as type : EPS file (*.eps) =
Save	<ctrl>-S</ctrl>	Selection
Save <u>A</u> s		test.eps
Export Setup		
Print Pre <u>v</u> iew		Save Filter Cancel
Print	<ctrl>-P</ctrl>	

Figure 7.7: "Save As ..." menu (left) and Matlab file browser (right). A number of image file formats like pdf, eps, fig, etc. can be generated with the Matlab "Save as ..." functionality.

### 7.3 Types of Data Views

By default ezplot displays uncalibrated eye position data of the left eye in units of pixels over time. This default option is denoted with "LeftEyeRaw" in the "Plot" menu. The right eye data can be plotted with the corresponding menu entries with a "Right" prefix.

There are also other ways to present the data that is contained in the EYESEECAM data file. Instead of the raw data the calibrated eye positions can also be plotted with the "LeftEyePos" or "RightEyePos" menus. The eye movement velocities can be plotted with the "LeftEyeVel" or "RightEyeVel" menus. Another means to present the data is not by plotting it over time but by plotting the vertical over the horizontal component. Such an example is given in Figure 7.8.



Figure 7.8: Calibrated eye position data with the vertical component plotted over the horizontal component (left subplot in right figure). Eye movement data can not only be plotted over time as in Figure 7.3 but also with one of the horizontal, vertical, or torsionl components plotted over another component. This plotting function can be activated with the "Plot" menu entry "LeftEye3D" (left).

The Matlab plotting commands are directly integrated into the EYESEECAM data files. These files contain a Matlab string variable called "EvalEZPlot" in which the plotting program is stored. This string can be viewed either with Matlab, with <code>ezplot</code> or with <code>octave<sup>2</sup></code>. After loading an EYESEECAM data file into the Matlab workspace the command <code>eval(EvalEZPlot)</code> would generate exactly the plot from Figure 7.3. The format of the EYESEECAM data file is explained in Chapter F.1 and in the next chapter some restricted methods of using <code>ezplot</code> for generating customized plots are discussed.

 $<sup>^{2}</sup>$ octave usage is shortly explained in Chapter 7.5.

## 7.4 Matlab Command Window

Advanced users who know how to use Matlab can work with a small subset of additional Matlab possibilities by invoking "Command ..." from the "Actions" menu of ezplot. The Matlab stand-alone program ezplot always has an accompanying xterm shell window open in which a prompt ">>" appears as soon as "Command ..." was invoked from the menu (see Figure 7.9). Some of the simpler Matlab commands can be entered at the prompt in this xterm window. It is possible to create an additional figure by entering figure (2), for example. The Matlab command plot can then be used to create customized plots. Invoking the command eval (EvalEZPlot) would simply generate again the plot from Figure 7.3, however, more sophisticated plots are also possible.



Figure 7.9: The Matlab stand-alone program is executed in an xterm window (right). As soon as the "Command ..." menu is activated (left) a prompt ">>" appears in the xterm window and a restricted subset of simple Matlab commands can be entered.

## 7.5 Data Browsing with Octave

ezplot is a stand-alone Matlab application, however, compared to Matlab its functionality is drastically reduced. An alternative way to browse the Matlab data files that are generated by EYESEECAM and to generate plots is octave. octave is a free program that clones part of the Matlab functionality. While Matlab is not shipped with EYESEECAM due to its high price octave is shipped together with EYESEECAM. octave can be invoked either from the desktop by clicking on the octave icon or from the shell by entering octave. A small octave session for plotting the horizontal eye position is displayed in Figure 7.10. octave uses gnuplot as the plotting program. Further documentation of these two programs can be obtained from the respective project websites.



Figure 7.10: The free Matlab clone octave can be used for generating simple plots of the EYESEECAM data. In this example a data file is loaded with the function load. The warnings displayed by octave can be ignored. The data file contains a Matlab string with the name EvalDataColumns. When eval (EvalDataColumns) is invoked a structure col is generated whose fields contain the column indices of the Data matrix. The command plot (Data(:,col.LeftEyeHor)) generates a plot of the calibrated eye position data.

# 8 Sequencial Control

8.1 Introductions to Profiles

??

- 8.2 Receivers
- 8.2.1 Diffraction Pattern

??

## 9 Database

EYESEECAM contains some basic database functionality which can be accessed from the "Database" tab. The database consists of a number of tables and each of these tables provides support for managing one of the following items:

- **Patients** Patient data like names, birthdays, etc. can be entered and edited (see figure 9.1)
- **Examinations** Examinations like caloric stimulation, saccade metrics, etc. can be selected from the predefined entries. New examinations can also be created and edited.
- **Conditions** Conditions like light, darkness, control, etc. can be selected from the predefined entries New conditions can also be created and edited.
- **Measurements** Measurements are in a 1:n relation to Examinations, i.e., a certain examination can contain n measurements. A caloric stimulation, for example, consists of five individual measurements. After a new examination is created at least one measurement must be added to the Measurements table. A certain measurement is either defined by a measurement time or by a (stimulation) profile.
- **Trials** Trials are in a 1:n relation to Measurements, i.e., a certain measurement can contain n trials. If everithing goes well during a measurement only one item will be added automatically to the Trials table. If something has gone wrong the measurement can be repeated and the table will be extended by an additional item.
- **Analysis** Analyses are in a 1:n relation to Examinations, i.e, a certain examination can contain n analyses. The caloric stimulation, for example, can be analyzed by the predefined stand alone Matlab application or by a new user defined function.
- **Examinator** Examinator name and contact data are managed in this table.
- System System settings, like paths to input and output data are managed here.

The database supports the user in carrying out the required examinations. Baiscally, the user simply has to add a new patient and enter his data and then select the desired examination from the Examinations tab, e.g., the caloric stimulation. After this step he can click on the Measurements tab and select the measurement he wants to perform next, e.g., the cold water stimulation of the left ear. Now the examination can be started by clicking on the "Eyetracker" tab and by subsequently pressing the "Start" button in the "Record" tab. Before the button is pressed the proper eye alignment should be monitored.

After data acquisition is finished a new file will be saved to disk. The name of this new file is assembled from the item IDs that were previously selected on the "Database" tab.

Immediately after the file is stored its contents can be viewed by clicking on the "Plot ...." button of the "Record" tab (see Chapter 7 for details). Another way to view (previously) recorded data is to click again on the "Database" tab. There, the desired trial can be selected from the "Trials" tab. A press on the "Plot Data" button, will start the data viewer.

eyese	e-vog								_ □
ile <u>E</u> ye	Tracking <u>S</u> wit	ches <u>S</u> ettings	s <u>H</u> elp						
ye Track	er Database	All Camera	as						
		>		Inse	rt New				
Patient	Examination	Condition	Measurement	Trials	Analysis	System	Examinato	r	
ID		1							
Name	9	Mustermann	I						
First I	Name	John							
Addre	ess	Munich							
Sex		M							
RegN	Ir	1							
Diagr	nosis								
Birtho	lav	2001							
		2001							
ID	Name	First Nan	ne Address	S	x Reg	Nr Dia	gnosis	Birthday	-
1 1	Mustermann	John	Munich	М	1			2001	

Figure 9.1: The "Patient" table of the database.

## 9.1 Database Tab

On the "Database" tab regular database operations can be performed. These operations are common to all database applications and include forward and backward navigations as well as the creation, editing, and deletion of items (rows) in a database table. All these actions can be accessed from the button bar above the tab area.

**Bar with buttons** The arrow buttons "<" and ">" switch between different entries of the table. The buttons "Insert New", "Update" and "Delete" change the content of the table. Update

Delete

**Tabs** with the names of the tables.

< >



Insert New

**Data form** In this form you can edit the content of the table. Some entries or given fields which stand for a distinct function of the database are marked grey and cannot be edited.

ID	1	
Examination	Kalorik	id 1
Name	Kalibration	
Profile	calib.mat	
Duration (sec)	45	

**Table-view** This table shows clearly arranged all entries inside the current table.

	ID	Examination	id	Name	Profile	Duration (sec)	l l
1	1	Kalorik	1	Kalibration	calib.mat	45	
2	2	Kalorik	1	Referenz	record.120s.mat	120	
3	3	Kalorik	1	Rechts warm	record.120s.mat	120	
4	4	Kalorik	1	Links warm	record.120s.mat	120	
5	5	Kalorik	1	Links kalt	record.120s.mat	120	
6	6	Kalorik	1	Rechts kalt	record.120s.mat	120	
							í l
							E
							•

## 9.2 Usage of the database

### 9.2.1 Patient

In this example a new file for a patient named "John Newman" will be created. With the key "Insert New" an empty entry is generated. Now you can fill in the empty fields with the patient's data. To save the edited patient form in the database click the button "Update". Any further changes of this form can be saved with the "Update" function. See figure 9.2.

	racking Switc	has Sattings H	aln					
e <u>L</u> yei			eib					
e nacke	Database	AirCairleias						
	<			Insert N	lew	Update	Delete	
Patient	Examination	Condition Me	asurement	Trials Ana	alysis Sy	/stem Examina	ator	
ID		2						
Name		Newmann						
First №	lame	John						
Addre	SS	Munich						
Sex		М						
RegNr		2						
Diagn	osis							
Birthda	ау	01.01.2007						
					Deable	Diagnosis	Birthday	
ID	Name	First Name	Address	Sex	Regivi	a		

Figure 9.2: Create new patient.

#### 9.2.2 Examination

You can find predetermined examinations in the table called "Examination", see figure 9.3. You can also add your own or new examinations. How you add the examinations is the same way as explained in chapter "Patient".

X eyesee.vog								-	- 🗆 🗙
Eile Eye Tracking Swite	ches <u>S</u> ettings <u>H</u> e	lp							
Eye Tracker Database	All Cameras								
	• >		Ins	ert Ne	W	) <u> </u>	Jpdate	Delet	te
Patient Examination	Condition Mea	asurement	Trials	Analy	/sis	System	Examinator		
ID	2								
Name	Newmann								
First Name	John								
Address	Munich								
Sex	М								
RegNr	2								
Diagnosis									
Birthday	01.01.2007								
ID Name	First Name	Address	9	ex	RegNr	Dia	gnosis	Birthday	
1 1 Mustermann	John	Munich	N	1	1			2001	_
2 2									
									-
Initialized 1 camera.									

Figure 9.3: Table Examination.

#### 9.2.3 Measurement

The examination consists of one or more measurements. Figure 9.4 shows all different measurements of the examination "Kalorik". Each new defined examination gets automatically a default measurement named calibration.

In figure 9.5 you can see an example of a newly defined measurement "My\_Measurement". You can also add your own or new measurements.

There is one important aspect of the Measurements table: In the column "Profile" either the duration<sup>1</sup> in seconds of the measurement should be entered here or the name of a Matlab version 4 data file. This data file must adhere to a specific syntax that is described in more detail in Section G.1 on page 77. It defines the duration and a synchronized sequence of stimulations that are applied as soon as the measurement is started. Possible stimulations are a visual stimulus like an optokinetic motion pattern presented on a video beamer or a vestibular galvanic current that is controlled by the analog outputs of connected data aquisition hardware like USB-DUX.

<sup>&</sup>lt;sup>1</sup>Using a duration instead of a stimulation profile is the easiest way to operate EyeSeeCam. If the user wants to deliberately start and stop measurements without the restrictions of a predefined duration he can enter a ridiculously high value for the duration and press the "Start" and "Stop" buttons on the "Eyetracker/Record" tab whenever he wishes.

ve T	Frack	er Databas	e All Can	neras	5			
						Insert New		
Pati	ient	Examination	Measur	emen	nt Trials Analysi	s System		
10	D		1					
E	Exami	ination	Kalorik			id 1		
Ν	Vame		Kalibrati	on				
F	Profile	<u>.</u>	calib ma	đ				
F	Profile Durati	on (sec)	calib.ma	t				
F	Profile Durati	on (sec)	calib.ma	it			Load Meas	urement
F	Profile Ourati	on (sec)	calib.ma 45	id	Name	Profile	Load Meas	urement
F C	Profile Durati	on (sec) Examination Kalorik	calib.ma 45	id 1	Name Kalibration	Profile	Load Meas Duration (sec)	urement
F C 1 2	Profile Durati	on (sec) Examinatic Kalorik Kalorik	calib.ma 45	id 1	Name Kalibration Referenz	Profile calib.mat record.120s.mat	Load Meas Duration (sec) 45 120	urement
F [ 1 2 3	Profile Durati ID 1 2 3	on (sec) Examinatio Kalorik Kalorik Kalorik	calib.ma	id 1 1	Name Kalibration Referenz Rechts warm	Profile calib.mat record 120s.mat record 120s.mat	Load Meas Duration (sec) 45 120 120	urement
F 1 2 3 4	Profile Durati	e on (sec) Examination Kalorik Kalorik Kalorik Kalorik	calib.ma 45	id 1 1 1 1	Name Kalibration Referenz Rechts warm Links warm	Profile calib mat record 120s mat record 120s mat record 120s mat	Load Meas Duration (sec) 45 120 120 120	urement
F 1 2 3 4 5	Profile Durati	e on (sec) Examination Kalorik Kalorik Kalorik Kalorik Kalorik	calib.ma	id 1 1 1 1 1	Name Kalibration Referenz Rechts warm Links warm Links kalt	Profile callb mat record 120s.mat record 120s.mat record 120s.mat record 120s.mat	Load Meas Duration (sec) 45 120 120 120 120 120	urement
F 1 2 3 4 5 6	Profile Durati 1 2 3 4 5 6	e on (sec) Examination Kalorik Kalorik Kalorik Kalorik Kalorik Kalorik	calib.ma	t 1 1 1 1 1 1 1 1 1 1	Name Kalibration Rechts warm Links warm Links kalt Rechts kalt	Profile calib.mat record 120s.mat record 120s.mat record 120s.mat record 120s.mat	Load Meas 45 120 120 120 120 120 120 120	urement

Figure 9.4: Table Measurement.

The directory from which these stimulation profile data files are loaded is defined in the database table System (see Section 9.2.9 on page 50 for details).

X eyesee-vog					_ <b>□</b> ×
<u>Eile</u> <u>Eye</u> Tracking <u>S</u> witch	nes <u>S</u> ettings <u>H</u> el	р			
Eye Tracker Database	All Cameras				
	>	In	sert New	Update	Delete
Patient Examination	Measurement	Trials Analysis	System		]
ID	19				
Examination	My_Examination		id 7		
Name	My_Measurement				
Profile	record.30s.mat				
Duration (sec)					
				Loa	d Measurement
ID Examination	id Na	me	Profile	Duration (	sec)
1 19 My Examinati	on  7  My	Measurement	record.30s.mat		
					* *

Figure 9.5: New measurement.

#### 9.2.4 Loading Measurement

After you have chosen a measurement you load it with the key "Load Measurement". The program automatically will switch to the tab "Eye Tracker". You start the measurement by clicking the key "Start Record". The duration of the measurement is determined by the previously loaded file "profile.mat" (see chapter "Measurement").

#### 9.2.5 Measurement Results : Trials

The content of the table "Trials" consists of the automatically generated filenames of all measurements of the previously chosen examinations. See figure 9.6.

еп	racke			annorao					
		<	>	]					
ati	ent	Examination	Meas	urement Trials	Analys	sis System			
IC	)		1						
Р	atien	t	Muster	mann		id 1			
F	xamir	nation	Kaloril	/		id 1	5		
-		nauon	I Aloni						
			_						
Μ	1easu	irement	Refere	enz		id 2			
M Fi	1easu ilenar	irement me	Refere	nz u001_m002_mv1		id 2			
M Fi	ilenar ilenar	irement me	Refere	nz u001_m002_mv1		id 2			Plot data
F	ilenar ilenar	irement me	Refere	u001_m002_mv1		id 2		[	Plot data
M Fi	ilenar ilenar ielect	rement me Patient	Refere	u001_m002_mv1	id	id 2	id	Filename	Plot data Select
M FI S	1easu ilenar ielect	Patient Mustermann	Refere	u001_m002_mv1	id 1	id 2	id 2	Filename poot_uoo1_m002_mv/	Plot data Select
M F S	1easu ilenar ielect 1D 1 2 3	Patient Mustermann Mustermann	Refere p001_ x id 1 1	Examination kalorik Kalorik	id 1 1	Id 2 Measurement Referenz Rechts warm	id 2 3 4	Filename p001_u001_m002_mv p001_u001_m003_mv	Plot data Select
M F S 1 2 3 4	1easu ilenar ielect 1 2 3 4	Patient Mustermann Mustermann Mustermann Mustermann	Refere p001_ x id 1 1 1 1	Examination Kalorik Kalorik Kalorik	id 1 1 1	id 2 Measurement Referenz Rechts warm Links warm Links kait	id 2 3 4 5	Filename p001_u001_m002_mv p001_u001_m003_mv p001_u001_m003_mv p001_u001_m003_mv	Plot data           Select           1           2           3           4
M F S 1 2 3 4 5	IEasu ilenar ielect 1 2 3 4 5	Patient Mustermann Mustermann Mustermann Mustermann	Refere p001_ x id 1 1 1 1 1 1 1	Examination Kalorik Kalorik Kalorik Kalorik	id 1 1 1 1 1	id 2 Measurement Referenz Rechts warm Links kalt Rechts kalt	id 2 3 4 5 6	Filename <b>pool_uool_moo2_mv</b> pool_uool_moo3_mv pool_uool_moo5_mv	Plot data Select I I 2 1 3 1 4 1 5 1

Figure 9.6: Table Trials.

The analysis of a trials (results of a measurement) is based on only one successful trial, therefore you can mark the valid trial in the table "Trials". You actually mark the trial in the column "Valid". As a default adjustment the last trial is marked as valid. You can also mark an trial afterwards as valid.

Figure 9.7 shows a repetition of the measurement "Referenz" with the last trial marked as valid.

	ID	Patient	id	Examination	id	Condition	id	Measurement	id	Filename	Valid	-
1	1	Mustermann		Kalorik	1	default	1	Referenz		p0001e01c01m02t1	0	
2	2	Mustermann	1	Kalorik	1	default	1	Rechts warm	3	p0001e01c01m03t2	1	
3	3	Mustermann	1	Kalorik	1	default	1	Links warm	4	p0001e01c01m04t3	1	
4	4	Mustermann	1	Kalorik	1	default	1	Links kalt	5	p0001e01c01m05t4	1	
5	5	Mustermann	1	Kalorik	1	default	1	Rechts kalt	6	p0001e01c01m06t5	1	L
6	6	Mustermann	1	Kalorik	1	default	1	Referenz	2	p0001e01c01m02t6	1	+

Figure 9.7: Table Trials.

#### 9.2.6 File Names

The file names assembled by the database have the following structure: pXXXXeXXcXXmXXtXX.mat. An example of a file name is p0001e03c00m02t02.mat. The first letter of the file name p prefixes the patient ID, which in this example is 1. The examination is denoted with the letter e which is followed by the examination ID. Condition, Measurement, and Trial are denoted by the letters c, m, and t, respectively. Although with this naming convention file names tend to get quite long and cryptical, this approach can be quite helpful in the automated analyisis of huge data sets with many patients and conditions. The thus created and named files are stored to a location that is defined in the table "System" (see Section 9.2.9). The "System" entry with the name "data" contains the storage path in its "Value" field.

#### 9.2.7 Plotting data

To create a figure of your collected data you mark a results file out of the table "Trials" and then click the button "Plot Data".

#### 9.2.8 Data analysis

In the table "Analysis" you can find all possible analysis programs for your selected examination, see figure 9.8. These programs are in a special directory (see chapter "System").

eyesee-vog
ile Eye Tracking Switches Settings Help
Eye Tracker Database All Cameras
Insert New         Update         Delete
Patient Examination Measurement Trials Analysis System
ID 1
Examination Kalorik id 1
Script Name kalorik
Data analysis
ID Examination id Script Name
1 1 Kalorik 1 kalorik
nitialized 1 camera.

Figure 9.8: Table Analysis.

This figure shows a selection of the analysis program "Kalorik". This program analyses all previously selected trials from the table "Trials". The results of the analysis are plotted in a figure placed in the new window.

#### 9.2.9 System

System-specific adjustments can be done in the table "System", see figure 9.9.

ey	esee	vog								-		2
e <u>E</u>	Еуе Т	racking <u>S</u> witc	hes <u>S</u> ettings	Help								
/e Ti	racke	r Database	All Camera	s								
		<										
Pati	ent	Examination	Condition	Measurement	Trials	Analysis	System	Examinator				
IC N	) ame		1 data									
V	alue		/home/voggu	est/eyeseecam	/data/			Change	Directory			
	ID	Name			∨alue						-	]
1	ID 1	Name data			Value /home/vo	igguest/eyes	seecam/da	ta/			-	
<b>1</b> 2	ID 1 2	Name data profile			∨alue /home/vo /home/vo	igguest/eyes	seecam/da seecam/pr	ta/ ofile/			-	
1 2 3	ID 1 2 3	Name data profile analysis			∨alue /home/vo /home/vo /home/vo	gguest/eyes gguest/eyes gguest/eyes	seecam/da seecam/pri seecam/an	ta/ ofile/ alysis/			-	
1 2 3 4	ID 1 2 3 4	Name data profile analysis display width			Value /home/vo /home/vo /home/vo 40	igguest/eyes igguest/eyes igguest/eyes	seecam/da seecam/pr seecam/an	ta/ ofile/ alysis/				
1 2 3 4 5	ID 1 2 3 4 5	Name data profile analysis display width display distan	ce		Value /home/vo /home/vo /home/vo 40 100	igguest/eyes igguest/eyes igguest/eyes	seecam/da seecam/pr seecam/an	ta/ ofile/ alysis/				

Figure 9.9: Table System.

The entries "profile" and "analysis" define the directories from where the profile files or analysis programs will be loaded. The entry "data" defines the directory where your data has been or is going to be saved.

#### 9.2.10 Examinator

Here you can add the name and additional information of the examiner, see figure 9.10.

cyese	ee.vu	9								
e <u>E</u> ye	e Trac	king <u>S</u> witc	hes <u>S</u> ettings	Help						
e Trac	ker	Database	All Camera	IS						
		<			Insert New		Update		Delete	
Patient	E	camination	Condition	Measurement	Trials Analysis	System	Examinato	or		
ID			1							
Nam	ne		Musterdoc							
First	Nam	e	John							
Contact 1111										
10	1 C	lame			First Name			Contact	(	•
1 1	N	lusterdoc			John			1111		
•									••	4 •

Figure 9.10: Table Examinator.

# Part III

# **Reference Manual**

## A Troubleshooting

## A.1 MacBook

**Computer hangs during boot sequence** The computer has been switched on or has just been re-booted. White text has been prompted on a black background screen. There has not been progress for a while.

Press and hold the power button until the computer is shut down. Wait a few seconds. Turn the computer on again. In case the problem remains, plug off any periphery, e.g. mouse, cameras.

**Computer hangs during shut down** Press and hold the power button until the computer is shut down.

PC	MacBook	PC	MacBook	PC	MacBook
Alt	Apple	[	Apple 5	<	^
Alt Gr	Alt	]	Apple 6	>	0
@	Apple L	Ι	Apple 7	^	<
~	Apple +	{	Apple 8	0	>
Del	(missing)	}	Apple 9		

**Missing keys** The MacBook's keyboard differs from the common PC keyboards.

## A.2 Cameras

**No IEEE1394 ("FireWire") cameras detected** Connect the IEEE1394 cable. Wait for 2 seconds, allow the camera to initialize. Then start EyeSeeCam. If no IEEE1394 cameras is found,

- most likely there is already an instance of EyeSeeCam program running. Press Ctrl+Esc. A window named ProcessTable appears. Look for "eyeseecam", select it and press button Kill. Quit the ProcessTable.
- 2. there might be a problem with the kernel modules that have "1394" in their names. We have observed such problems when un-plugging a IEEE1394 camera while EyeSeeCam is running. Un-plug the cameras and reboot the computer. <sup>1</sup>

#### Lost frames ??

<sup>&</sup>lt;sup>1</sup>Advanced users: unload and reload those modules using lsmod, rmmod and modprob.

## B Eye Movements in 3D with Scleral Markers

For measuring 3D eye movements that include torsional rotations around the line of sight EYE-SEECAM currently requires dark artificial landmarks that can be applied to the sclera with a cosmetic pigment and a sterile surgical pen. The main advantage of using such artificial landmarks is the superior data quality in comparison to measurements that are based on natural iris landmarks. However, the procedure of applying the scleral markers can be called invasive, although it is tolerated very well. This chapter describes how the scleral markers can be applied.

## **B.1 Marker Application**

This chapter lists both the material that is required for applying the dark artificial markers to the sclera and the single steps of the marker application process. The markers consist of a dark pigment that –being a cosmetic product– is already tested for physiological compatibility. The pigment absorbs infrared light and, therefore, the markers appear as dark or almost black dots with a high contrast to the white sclera. The markers have to be applied with a sterile surgical pen onto the anesthetized sclera.

#### Material

- 1. Dark cosmetic pigment<sup>1</sup>
- 2. Contact lens holder; in the one half the pigment is kept and in the other half it can be dissolved with saline solution
- 3. Conjuncain EDO 0,4% eye drops
- 4. Disposable injection with a small needle
- 5. Physiological saline solution; this can be kept in the injection.
- 6. Sterile surgical pen<sup>2</sup>

<sup>&</sup>lt;sup>1</sup>The dark cosmetic pigment can be purchased from www.eyeseecam.com <sup>2</sup>Surgical pen, Fine, 26.665.00.000 from Pfm AG in Cologne, Germany, http://www.pfm-ag.de/

#### Process

- 1. The eye needs to be anesthetized with a drop of Conjucain about 3 minutes before the markers can be applied to the sclera.
- 2. A pasty solution can be dissolved in the contact lens holder from a small amount of pigment (as much as fits on the tip of a scalpel) and a much smaller drop of physiological saline solution. The thin injection needle can be used for the drop.
- 3. The surgical pen can then be permeated with this pasty solution.
- 4. The most *important step* consists of painting the markers on the sclera as soon as the eye anesthesia is in effect.
  - a) The subject / patient has to assume the same head and trunk position as during the later examination, since the head orientation can have an effect on the eye position.
  - b) The subject / patient has to assume an excentric eye position by fixating an excentric dot. The sclera can then be approached with the pen from the opposite side. The patient must not blink and keep the eye open for about half a minute until the pigment is dry, otherwise the marker may be washed out by the eyelid.
  - c) The markers have to be painted at a distance of about 1-2 mm from the limbus.
  - d) There are many methods around of how to apply the markers, probably as many as there are examinators. One of the methods is to repeatedly paint the dot with a gentle pressure on the sclera.
- 5. This process can be applied more than one time if the marker is washed out. Preferably, the same location should be chosen on the sclera for the new marker.
- 6. Finally, the EYESEECAM goggles can be put on and the appearance of the eye and the markers can be monitored in the EYESEECAM video displays.

This process has to be practiced. After the first few trials the markers will probably wash out in minutes, however, after some practice the markers will stay for at least one hour and the time for completing the process can be reduced to less than ten minutes. The subsequent steps that are required to prepare the software for marker tracking are detailed in Chapter. The tracked marker positions are saved to a Matlab data file whose format is described in Chapter F.1.

### **B.2 Detection**

This section describes how to work with EyeSeeCam in order to detect the sclera markers and evaluate the torsional eye position.

1. The checkbox "Markers" in buttongroup "Enable" in tab "Eye Tracker/Common" must be checked. The default state of this checkbox can be set in camera.xml, see section E.1, camera/imgproc/marker\_enable.

- 2. Perform a successful gaze calibration.
- 3. Switch Laser Pattern on again. Advice the subject to fixate on the center laser dot. Horizontal and vertical eye position are close to zero now.
- 4. Select the tab "Left", "Right" or "Binocular", if the subject has the markers in the left eye, the right eye or both eyes, respectively.
- 5. Enlarge the videobox (by double-click or mouse wheel). Make sure that item "Markers" of the videobox'es popup-menu is checked, so yellow vectors will be diplayed from pupile to markers once they are detected. We also recommed to check item "Globe".
- 6. Press and hold Ctrl+Alt (on MacBooks: Ctrl+Apple) and draw a rectangle with the left mouse button over the two markers.
  - The entire marker must be *within* the rectangle. The *border* of the rectangle should cover the evenly white sclera, and not be disturbed by any high-contrast details.
  - You may repeat the rectangle-drawing until it fits.
  - A single click with the left mouse button deletes the marker (remember to press and hold Ctrl+Alt/Apple, else the left mouse button has no effect). The rectangle is interpreted by the program as marker sample pattern if it is eitherr in the left or right half of the image. If you happen to draw the rectangle over the vertical mid-line, it is interpreted as the new ROI (region of interest for pupil detection), thereafter you must set the desired ROI again.
  - If both eyes have sclera markers, repeat this procedure for the other eye.
- 7. Check the range of eye positions where the marker detection works well: Switch off the laser pattern and advice the subject to look around, first with head upright, then with head tilted to the left (left ear to left shoulder), then to the right.
  - Watch for interactions of the markers to the eye lids. TROUBLESHOOTING: If the rectangle was drawn too large it gets in touch with the eye lid too early. Re-draw the rectangle a little smaller.
  - Watch for interactions of the markers to illumination reflexes. TROUBLESHOOTING: Try to cover the corresponding illumination LED, which is embedded in the goggle's frame, e.g. by black tape.
  - Watch the Globe (dark blue spherical wire frame) when the eye moves. The Globe should *stick* to the markers!
    - TROUBLESHOOTING: Do the calibration again.
- 8. Settings: Select the eye tab "Left" or "Right" and watch the graph "T" (torsional component) in plot "Postion" for discontinuities.
  - Slider "Search range" on tab "Marker" controls the angular length of the search path. The search path is displayed as a yellow arc in the videobox. Do not set a value that is much bigger than required. We recommend about 3° more than the expected maximum amplitude of ocular torsion. The measurement is easily disturbed if the search path touches the eye lids.

- Slider "Pixel noise": You might only want to fine-tune this parameter for markers that are about to disappear (e.g. one hour after application). Watch the sample images in the corners of the videobox to see the effect of this value. The background must appear white, not gray and dirty.
- 9. When recording, the columns about markers will be included in the MAT-file and TXT-file if "Markers" are enabled (checkbox on tab "Common") and the rectangle for marker detection has been drawn.

## C Tracking of Horizontal, Vertical, and Torsional Eye Movements

Methods and corresponding bibliography from

T. Dera, G. Böning, S. Bardins, and E. Schneider, "Low-latency video tracking of horizontal, vertical, and torsional eye movements as a basis for 3DOF realtime motion control of a head-mounted camera," in *Proceedings of the IEEE Conference on Systems, Man and Cybernetics (SMC2006)*, Taipei, Taiwan, 2006.

## C.1 METHODS

#### C.1.1 Overview

The evaluation of eye orientation in the head involves two major steps.

- 1. The pupil in the image is detected, in both horizontal and vertical directions.
- 2. Artificial sclera markers are used to determine the torsional component of eye orientation, i.e., the rotation angle around the axis of gaze direction.

The second step, the detection of the landmarks on the eyeball even for eccentric eye orientations, requires a geometric model of the eyeball. The geometric model basically is a transformation function that provides the functionality listed in table C.1.

Input	Output
Image coordinate	Eye orientation, zero
	torsion
Eye orientation	Image coordinate of
	pupil
Image coordinate, eye	Location of marker on
orientation	the eyeball
Location of a marker on	Image coordinate of the
the eyeball, eye orienta-	marker
tion	
Sequence of pupil im-	no output, but internal
age coordinates during	parameters adjusted.
calibration	

Table C.1: Interfaces of the geometric model.

Figure C.1: Left: A cutout of the VOG video image showing just the pupil. Although the pupil area may be partially covered by light reflexes, the detection of the pupil center should not be affected. Middle: The pupil edge is detected. Right: Edge segments that do not meet the expected edge curvature of the elliptic pupil edge are removed. Fitting an ellipse to the remaining pixels results in the correct pupil center coordinates.

#### C.1.2 Pupil Detection

For the detection of the 2D projection of the pupil within a video image the parameters describing the pupil position and shape are initially estimated and then sequentially improved by several processing steps. All steps are based on the assumption that the pupil can be identified by its brightness and by the brightness gradient between the pupil and the iris or, e.g., reflections of the illumination.

Brightness thresholds are therefore automatically determined for each frame by analyzing its brightness histogram. These thresholds are utilized to form horizontal and vertical projections of the binarized image. After background subtraction, the back-projection of these profiles yields a first guess of the pupil's center coordinate. Starting from this center the connected cloud of pupil pixels is searched along a spiral path. A principle component analysis of the pixel cloud yields a first and robust estimate of the pupil's ellipse parameters, its center, the length of both axes, and its rotational angle.

To increase the accuracy of the 2D pupil detection, a further edge search and edge analysis following a method proposed in [1] is performed. Starting from the edge of the previously detected pixel cloud, a more precise edge search is performed using a 3x3 pixel mean filter until a connected edge around the pupil is found, followed by a curvature analysis which eliminates edge inhomogeneities. Finally, a 2D variance image of the pupil is calculated to allow the identification and exclusion of edge pixels with an unusually high variation in their 7x7 pixel neighborhood. This step is intended to delete edge pixels which have not been identified by the curvature analysis but still are affected by nearby cornea reflections. An ellipse is linearly fitted [2] on the remaining edge pixels to achieve the final ellipse parameters that describe the 2D projection of the pupil. Fig. C.1

shows a representative image of the pupil (left) and the connected pupil edge (middle). The boundaries of the cornea reflections are accurately removed by curvature and variance analysis (right figure) to improve the results of the final fit operation.

The two actuators that control pan and tilt of the scene camera take pupil coordinates as input without further geometrical interpretation. This is because the calibration procedure relates measured pupil positions to given servo commands. This procedure was described previously in [3] and in the companion paper [4].

#### C.1.3 3D Geometric Model Approach for Eyeball and Camera

The geometric model described here is a sequence of three transfer functions. Fig. C.2 presents a concise graph of the data flow.



Figure C.2: Transformation sequence from pupil image coordinates to eye orientation as used for gaze evaluation. For calibration the reverse transformation calculates expected pupil coordinates from given eye orientations.

The three transfer functions are each specialized on a certain task:

- 1. The affine transformation transforms image coordinates. The parameters of the internal matrix are set by the calibration procedure. See section C.1.6.
- 2. The axially symmetric geometric model is based on the dimensions of eyeball radius, camera to eyeball distance, camera focal length, etc. See section C.1.4.
- 3. Since another reference for gaze direction may be desired than the eyeball-camera-axis, a rotation is applied to the coordinate system. See section C.1.5.

#### C.1.4 Geometric Model: Axially symmetric configuration

In an ideal configuration, the eyeball is spherical and centered at the origin of the coordinate system, as illustrated in Fig. C.3.

Figure C.3: The geometric model of eyeball and camera. In this idealized configuration the camera looks directly at the eyeball center. For eccentric gaze direction the pupil center is projected to the camera image plane at coordinates distant from the image center. The model works best for a pupil that is located below the spherical eyeball surface and above the iris plane. Optical properties of the cornea are not considered further.

In this coordinate system the gaze angle is zero in case the eye is oriented towards the pupil.

The configuration of the eye and the camera is symmetrical to the axis through the eyeball center and the camera focal point. Thus, the gaze direction can be separated into two angles: deviation  $\vartheta$  from this axis and an azimuth angle  $\varphi$  that equals the polar angle of the projected pupil center in the image, measured from the image center. Given the gaze angles  $(\vartheta, \varphi)$ , the pupil center is projected onto the polar image coordinates  $(r', \varphi')$  as described by the function

$$\frac{r'}{f} = \frac{\sin(\vartheta)}{c/r - \cos(\vartheta)}, \quad \varphi' = \varphi \tag{C.1}$$

with f camera focal length, c distance of the camera focal point to eyeball center, and r distance of pupil center to eyeball center. Polar image coordinates are finally transformed into Cartesian coordinates (x', y').

The reverse transformation is needed to process the video image and calculate the gaze direction. Since (C.1) cannot be solved analytically for  $\vartheta$ , an approximation is used instead:

$$\frac{r'}{f} \approx a \sin(b \vartheta) \quad \leftrightarrow \quad \vartheta \approx \frac{1}{b} \arcsin\left(\frac{1}{a} \frac{r'}{f}\right)$$
 (C.2)

with the parameters

$$a = \frac{(c/r)\sin(\arccos(c/r))}{1 - (c/r)^2}, \quad b = \frac{1 + c/r}{\sin(\arccos(c/r))}.$$

In (C.1) and (C.2) r' share the same maximum value and the same first derivative for small  $\vartheta$ . The deviation of the approximation from the exact formula is below 0.8% of eyeball radius for  $\vartheta < 90^{\circ}$ .

In 3D eye tracking, eye orientation (also referred to as eye position) is commonly represented as a rotation vector, i.e., a vector that is parallel to the axis of rotation. In contrast to the quaternion vector used in [5], the absolute value of which is  $\sin(\alpha/2)$ , we used the "Euler-Rodrigues parameters", a 3-component vector with an absolute value  $\tan(\alpha/2)$ . This definition considerably simplifies the mathematics [6]. Once the gaze direction is available from the spherical coordinate angles  $\vartheta$  and  $\varphi$ , the corresponding rotation vector can be calculated. It transforms the position vector of the pupil center from the reference direction to the current gaze direction.

#### C.1.5 Geometric Model: Rotation of the Coordinate System

A laser mounted on the goggles provides a gaze direction that corresponds to a straight-ahead view. In Fig. ?? this laser is in the white housing, and has an attached diffraction grating, as in [7]. It projects a pattern of laser points with the center-point being the brightest. As laser and camera are both mounted on the same platform, the angular relation is constant for the system, and independent of the user. The central laser point also provides a common reference direction for both eyes.

The position of the camera is given as the one eye orientation (a rotation vector)  $\mathbf{r}_{RC}$  which lets the eye look directly into the camera. Combining this rotation with the eye orientation  $\mathbf{r}_{C}$  Figure C.4: Illustration of the effect of the affine transformation. Left: un-calibrated, right: with calibrated parameters. Scaling, rotating and shifting any pupil center coordinate is equivalent to transforming the entire image before pupil detection. In this way the main factors of static deviation can be compensated.

calculated above, is equivalent to a rotation of the coordinate system. Thus, the resulting eye orientation  $\mathbf{r}_R$  is relative to the reference orientation rather than to the eyeball-camera axis,

$$\mathbf{r}_{R} = \mathbf{r}_{RC} \circ \mathbf{r}_{C} = rac{\mathbf{r}_{RC} + \mathbf{r}_{C} + \mathbf{r}_{RC} \times \mathbf{r}_{C}}{1 - \mathbf{r}_{RC} \cdot \mathbf{r}_{C}}$$

with the operator "o" denoting the non-commutative combination of two rotations [8].

#### C.1.6 Geometric Model: A Calibrated Affine Transformation of the Image

To account for real-world variations between individual users as well as variances within single sessions, an affine transformation of image coordinates is introduced. The two transformations described above in C.1.4 and C.1.5 are not designed for automatic parameter adjustments.

From the camera's point of view variations of eye-to-goggle position (tilt and pan, cameramirror-eye distance) and also physiological eyeball diameter variations result in variations of horizontal and vertical image shift  $\Delta_x$  and  $\Delta_y$ , respectively, image rotation  $\alpha$ , and image scale s. We therefore applied an additional affine transformation

$$\left(\begin{array}{c} x' \end{array}\right) = \left(\begin{array}{c} s\cos\alpha \end{array}\right) \\ \left(\begin{array}{c} x'' \end{array}\right)$$

to compensate for shift, rotation and scale deviations (see Fig. C.4

for illustration), with x'' and y'' the coordinates taken directly from the image.

Four adaptive parameters in a transformation matrix with six elements mean that two types of transformations are not considered: shear and anisotropic scaling. With our hardware configuration and the mechanical constraints no variations of that type are expected.

A calibration procedure must be performed to adjust the parameters of the affine transformation. Before calibration the affine transformation equals identity, with scale s set to 1, and rotation  $\alpha$  and shifts  $\Delta_x$  and  $\Delta_y$  set to zero.

During the calibration procedure a laser diffraction pattern is presented to the user, projected on a wall straight-ahead [7]. It consists of a bright laser point in the center and four adjacent laser points (above, below, left and right) which have a known angular distance ( $8.5^{\circ}$  at  $\lambda = 833$  nm) to the center-point. Other less bright laser points are ignored. The user fixates each of these five laser points for about 5–6 seconds, in an arbitrary order. Subsequently the corresponding pupil positions are analyzed by a cluster algorithm [9], resulting in five representative pupil coordinates (average of cluster). Finally, the parameters (scale, rotate, horizontal and vertical shift) are calculated, transforming the cluster positions x'', y'' into the expected positions x', y'. The expected pupil positions were calculated earlier by simply applying the geometric model to the eye orientations that correspond to the given laser point.

#### C.1.7 Measuring Torsional Eye Movements

As many people do not display natural structures in their iris that have enough contrast for robust image processing, and because of the more simple search algorithm implementation, we use artificial pigment markers on the sclera (see Fig. C.5).

Figure C.5: Black pigment markers applied to the white sclera left and right of the iris.

Artificial markers have been used before [10], and the method is described in [11]. Other methods using contact lenses (e.g. [12]) are applied in clinical or research environments.

Once the pupil is found in the video image, the orientation of the eye in the head system is calculated by transfer functions C.1.6, C.1.4 and C.1.5 of the geometric model. The transformation used here was developed in order to avoid the complexity of a general matrix formulation.

Fig. C.5 shows an eye with two markers applied to the sclera. The markers consist of an infrared-absorbing cosmetic pigment. They were applied to the sclera by means of a sterile surgical pen. Prior to the application the pen was permeated with a solution prepared from a small amount of pigment and a drop of water [11].

The geometric model also allows detection of these sclera markers in case of eccentric view. When selecting the marker in the video by placing a rectangular field around it, zero torsion is assumed. The marker location on the eyeball is evaluated and stored. Then, for eccentric eye orientations and a requested torsional angle, the geometric model of the eyeball predicts the location of the marker. The pupil center is located closer to the eyeball center then the sclera markers. A difference of about 8% was found empirically. Further details of the the eye anatomy, e.g. the optical magnification effect of the cornea, are not taken into account.

A marker search is started in an angular range around zero torsion in clockwise and counterclockwise directions. The marker search has two phases.

- 1. Search along an arc-shaped search path to find the marker, with only pixel resolution.
- Center-of-mass analysis is performed on the region to calculate the marker position precisely.

Being aware of the restrictions of a real-time application, the following algorithms were developed as trade-offs for speed and precision, intentionally avoiding image filters, sorting, deep nested loops, etc.

Figure C.6: Marker detection and eye orientation. The *upper left image* illustrates the search path method. Copies of the selected marker fields are shown in the upper corners of the image. As the markers in this example have a bimodal brightness distribution, the signatures, displayed in the lower corners and on the search path, have double-stripes. The *upper right image* illustrates the center-of-mass evaluation. It shows the the search sample in the upper corners as well as the local sample on the search path. The markers appear bright and clear on black background. The *lower left and right images* illustrate eye orientations with ocular torsion.

Marker detection starts when the operator—with the video on the computer screen—uses the mouse to drag a rectangular region containing the dark marker and some "white" surrounding pixels of the sclera. Applying two markers, left and right of the iris, suppresses the noise from the pupil position in the final result of the torsional angle. The selected subimages are then stored, and they will serve as the search pattern for subsequent images. The geometric center of this subimage is fed into the calibrated geometric model, which returns locations in the eyeball coordinate system.

While the analysis of the search pattern is done only once, the following steps are applied to every frame:

- 1. Coarse search along a path
  - a) Image positions are calculated for zero torsion, and e.g.,  $12^{\circ}$  clockwise and  $12^{\circ}$  counterclockwise torsion. A circular arc is fitted onto these three image points. The arc-function is used as search path.
  - b) A signature s (one-dimensional array) is calculated along the search path, in the same way that a signature is calculated in the search pattern. Each element  $s_i$  of the signature array is calculated  $s_i = \max_j p_{ij} \min_j p_{ij}$ , with *i* a row index, *j* a column index, and  $p_{ij}$  the value of the pixel. Therefore, each signature element  $s_i$  expresses how much of a marker is found in row *i* (the pixel row is just as wide as the search sample), independent of the local brightness level of the sclera background.
  - c) The search pattern signature is compared row by row with the search path signature. The vertical position with the least squares residuals yields the first estimation of marker position.
- 2. Center-of-mass analysis
  - a) The region around the resulting position of the coarse search along the path is analyzed. Each pixel is scaled by a combined gamma and threshold function making background colors black (zero) and darker marker pixels bright.
  - b) Background is a polynomial function  $b(x, y) = b_1 + b_2 x + b_3 y + b_4 x y$  that fits best to the pixel values of the rectangle outline. With all the background pixels eliminated, only marker pixels contribute to the center-of-mass. Following this rule the center-of-mass position is calculated once for the search pattern.
  - c) Two image coordinates are of interest at this stage.
    - the center-of-mass where the marker was actually found, and
    - the center-of-mass of the search pattern moved to that position where it would have been in case of zero torsion.
  - d) Both center-of-mass image coordinates are transformed by the geometric model into locations in the eyeball coordinate system.
  - e) These two location are projected into the plane which is perpendicular to the gaze direction, and contains the eyeball center. The angle between the two corresponding rays from eyeball center to these projected locations yields the final result, the torsion angle.

## **D** User Environment

This chapter describes how the computer is pre-configured.

The root password is "EyeSee" by defaults.

## D.1 The User "eyesee"

Standard user is "eyesee". This user is logged in at startup without query for password. The *pass-word* is "EyeSee", same as the administrator's password. In the user's home folder /home/eyesee/ there exist two folders: .eyeseecam/ (the dot at the beginning makes this folder a hidden folder) and eyeseecam/.

Note that the files created by the application eyeseecam are owned by the user "root"<sup>1</sup> rather than the user "eyesee". Anybody has read and write permissions on those files. So the user "eyesee" is also allowed to make himself/herself the new owner.

## **D.2 Create a New Working Directory**

/home/eyesee/eyeseecam/ is intended to be the working directory. The work path of the desktop icon "EyeSeeCam" (Properties > Application > Work path) is set to this folder. In case you want to start eyeseecam from another working directory, edit the Properties > Application > Work path of a copy of the desktop icon appropriately. If your preferences differ from the default settings, you should also copy the configuration files \*.xml to the new working directory (see chapter **??**).

When eyeseecam starts up, it also looks in the hidden folder "~/.eyeseecam/" and the current directory for optional configuration files<sup>2</sup>.

## D.3 Create a New User

We have prepared the user eyesee to work with eyeseecam. In case you are an experienced administrator, you may want to create other users accounts. In order to run eyeseecam as another user,

• copy ~/.eyeseecam,

<sup>&</sup>lt;sup>1</sup>The application eyeseecam is owned by root and configured to run with superuser permissions. Consequently, the files created are owned by the superuser "root", no matter which user has called the application.

 $<sup>2^{2}</sup>$  ~/" is short for the current user's home folder.

- set CPU frequency to maximum ("Performance"), since eyeseecam is a real-time application and the user interface requires *idle* intervals,
- add the user to group "uucp" to permit access to serial ports.

## **E** Configuration Files

The files below control the defaults properties and parameter values of EyeSeeCam. They are not required. EyeSeeCam determines the default parameter values at program start in the following order, where the later overrules the current values:

- 1. the built-in values,
- 2. optional configuration files in the hidden folder ~/.eyeseecam/,
- 3. optional configuration files in the working folder (~/eyeseecam/ if you use the desktop link), and finally
- 4. the built-in system database.

The *system database* is queried with the results of the hardware detection procedure. It holds e.g. information about where and how an individual camera is mounted.

The resulting configuration is written to the hidden folder ~/.eyeseecam/ in several files with common suffix ".save.xml". You can use copies of these files where you remove the ".save" from the filename. Edit these copies in order to change the configuration. "KXMLEditor", "Conglomerate XML Editor" amd "Kate" are appropriated tools. XML contains a hierarchical structure of elements that look like <element attribute="value" />...</element> or <element attribute="value" />...</formallelement // It is easy to change a value. You may delete any element or attribute that you do not need.

Most important parameter is camera/grabber/hw\_pgr/height to set the camera speed: Reduce the height for more *frames per second* (*fps*).

### E.1 camera.xml

- camera/grabber/hw\_pgr/height (integer) The height of a given digital camera in pixels. There is a list of hw\_pgr elements, enumerated by the attribute id. id="0" refers to the first digital camera, id="1" to the second, and so on. Use this attribute to control the camera *speed*: A PointGrey FireFly MV runs with 120 frames per second at height="348", 131 frames per second at height="216", 300 fps at height="80", and 500 fps at height="40". IMPORTANT: Use the same value for all connected eye tracking cameras!
- **camera/imgproc/marker\_enable** (bool) Enable sclera marker detection by default. In GUItab EyeTracker/Common in buttongroup "Enable" the item "Markers" will be checked after program start.

## E.2 iofile.xml

??

## E.3 ports.xml

??

## E.4 servo.xml

??
# **F** Output Files

## F.1 Data in MAT-File

??

## F.2 Video in PGM-File

Multi-image Portable Graymap. The format specification is described at

http://netpbm.sourceforge.net/doc/pgm.html. EyeSeeCam saves images uncompressed, as they come from the camera, to file with the PGM frame header added.

Many programs process PGM, see

http://netpbm.sourceforge.net/doc/directory.html. There are even interpreters for Bayer tiled cameras:

http://netpbm.sourceforge.net/doc/pambayer.html.

Before July 2000, there could be at most one image in a PGM file. As a result, most tools to process PGM files ignore (and don't read) any data after the first image. Unfortunately, MATLAB R2006b is one of those ignorant tools.

Here is the (simplified) C source code that reads the meta information of a frame. See also section F.3 for the declaration of struct CAMERA\_property and TIMESTAMP. Finally "size" is set to the number of bytes of the image and the current file position is at the first byte of the image.

```
do // find a frame with requested camera id
{
 /**- move to next frame (no move in first loop with size==0) */
  error = fseek( iofile_pgm ->fp, size, SEEK_CUR );
  if (error) goto error_seek;
 /**- read magic number */
  read = fscanf( iofile_pgm->fp, " P%i ", &magic_number );
  if (read < 1) goto return_failure;
  switch (magic_number)
  {
   case 5: camera_property.color_mode = CAMERA_LUMINANCE; break;
   case 6: camera_property.color_mode = CAMERA_RGB; break;
    default: goto return_failure; break;
  }
 /**- default values in case no comment is given */
  cam_id = 0;
  memset( t, 0, sizeof(TIMESTAMP) );
  shift_h = shift_v = 0.;
  meta_imgproc.vog.found_pupil = false;
  meta_imgproc.vog.pupil_col = meta_imgproc.vog.pupil_row = 0.;
  meta_imgproc.eye_rotation[0] =
```

```
meta_imgproc.eye_rotation[1] =
  meta_imgproc.eye_rotation[2] = 0.;
  /**- read next character */
  while ((c = fgetc(iofile_pgm \rightarrow fp)) == (int) '#')
  {
    /**- read comment line */
     if (fgets( line_buffer, 80, iofile_pgm->fp ))
    /**- evaluate comment line with known keyword */
    sscanf( line_buffer , " %s", keyword );
if (strcmp( keyword, "cam" ) == 0)
sscanf( &line_buffer[3], " %i", &cam_id );
else if (strcmp( keyword, "time" ) == 0)
    cise if (stremp( keyword, time ) == 0)
sscanf( &line_buffer[4], " %li %li", &t->tv_sec, &t->tv_usec );
else if (stremp( keyword, "shift") == 0)
sscanf( &line_buffer[5], " %f %f", &shift_h, &shift_v );
else if (stremp( keyword, "pixelsize") == 0)
sscanf( &line_buffer[9], " %f %f",

                 &camera_property.pixelwidth,
     &camera_property.pixelheight );
else if (strcmp( keyword, "mirror" ) == 0)
sscanf( &line_buffer[6], " %i %i",
                  &camera_property.mirror_x ,
     &camera_property.mirror_y );
else if (strcmp( keyword, "vog" ) == 0)
     {
       meta_imgproc.vog.found_pupil = true;
       sscanf( &line_buffer[3], " %f %f ",
                  &meta_imgproc.vog.pupil_col,
                 &meta_imgproc.vog.pupil_row );
     }
     else if (strcmp( keyword, "eye" ) == 0)
sscanf( &line_buffer[3], " %lf %lf %lf ",
                 &meta_imgproc.eye_rotation[0],
                 &meta_imgproc.eye_rotation[1],
                  &meta_imgproc.eye_rotation[2] );
    }
  }
  /**- push back recent character which was not a comment */
  ungetc( c, iofile_pgm ->fp );
  /**- read image dimensions */
  read = fscanf( iofile_pgm ->fp, " %i %i %i ",
                      &camera_property.width, &camera_property.height, &maxval);
  if (read < 3) goto return_failure;
  /**- determine depth in byte per pixel */
  if (maxval > 0xff)
  {
     camera_property.color_mode = CAMERA_YUV422; //we do not have that with CAMERA_LUMINANCE
    camera_property.depth = 2;
  }
  else
    camera_property.depth = 1;
  size = camera_property.width * camera_property.height * camera_property.depth;
  /**- move to start of image, therefore skip a single whitespace */
  if (!isspace( fgetc( iofile_pgm->fp ))) goto return_failure;
} while (cam_id != track && //requested track not yet found
           !feof( iofile_pgm->fp )); //end of file not reached
```

#### F.3 Video in RAW-File

This file format comes without file header and frame headers. It's just the series of uncompressed images. The first pixels of each frame are sacrified to store some meta data. The advantage of \*.raw files is that (assuming there is only on camera or all cameras have the same image dimensions) having read the image size of the first frame(s), the positions of all frames in the file is known. This makes file access very easy and fast.

This is the current declaration of the data structure ("abstract"):

```
#ifdef WIN32 //-
#include <time.h>
typedef struct
{
  time_t tv_sec; /// < date and time (seconds)
  long tv_usec; /// < microseconds
TIMESTAMP;
_____
#include <sys/time.h>
typedef struct timeval TIMESTAMP;
#endif //WIN32 -
/** camera image properties at time of initialization */
struct CAMERA_property
{
  size_t width; /// < width of image in units of pixel</pre>
  size_t height; /// < height of image in units of pixel
  size_t depth; ///< depth of image in bytes per pixel</pre>
  enum CAMERA_colormode color_mode; /// < 0=grayscale, etc
  bool alternate; /// < both fields alternating into separate buffers, or interlaced?
  TIMESTAMP timestamp; /// < timestamp at camera initialization
  float pixelwidth; ///< width of a pixel (consider binning mode!), in units of mm float pixelheight; ///< height of a pixel (consider binning mode!), in units of mm
  int mirror_x; /// < factor +1 (no mirror) or -1 (mirror flips view horizontally)
  int mirror_y; /// < factor +1 (no mirror) or -1 (mirror flips view vertically)
  float frameperiod; /// < measured average frameperiod, in units of s
}:
/** image meta information */
struct IOFILE_RAW_abstract
{
  size_t sizeof_ABSTRACT;
  int track; //camera index
  /* image properties */
  struct CAMERA_property camera_property;
  /* VOG properties */
  bool vog;
  float x_deg;
  float y_deg;
  float x_pupil;
  float y_pupil;
  float latency;
} __attribute__((packed));
```

Here is the (simplified) C source code that reads the meta information from a frame:

do

```
{
 /**- move to next frame (no move in first loop with size==0) */
  error = fseek( iofile_raw ->fp, size, SEEK_CUR );
  if (error)
    goto return_failure;
 /**- read abstract from file */
  read = iofile_read( &abstract, sizeof( struct IOFILE_RAW_abstract ), iofile_raw ->fp );
  if (read < 1)
    goto error_read_abstract;
 /**- check if versions match */
  if (abstract.sizeof_ABSTRACT != sizeof( struct IOFILE_RAW_abstract ))
    goto error_size_of_abstract;
 /**- only images larger than their included abstract make sense */
  size = abstract.camera_property.width *
  abstract.camera_property.height *
  abstract.camera_property.depth;
  if (size < abstract.sizeof_ABSTRACT)
    goto error_size_of_image;
 /**- move back to the start of the abstract, which is also the start of the image frame */
  error = fseek( iofile_raw ->fp, -1 * abstract.sizeof_ABSTRACT, SEEK_CUR );
  if (error && !feof( iofile_raw ->fp ))
    goto return failure;
} while (abstract.track != track && //requested track not yet found
         !feof( iofile_raw ->fp )); //end of file not reached
```

## F.4 Video in DV-File

Videos from a digital camcorder that is connected via FireWire are transmitted in compressed DV format. Choose the \*.dv format if you want to store such a videostream to file. The videoframes are recorded without re-coding. WARNING: Do not choose this file format for any other video soure; this would result in *empty* target files without further notice!

This version of the DV file format is similar to the AVI Type 1. While AVI Type 1 has a file header and frame headers according to the RIFF specification, the DV format is just a serie of equal sized blocks (PAL 144000 bytes, NTSC 120000 bytes per frame). For Mac OS it is a well supported format, whereas many Windows tools prefer AVI Type 1 over pure DV.

## F.5 Video in AVI-File

This format is for lossy encoded videos. The video files are much smaller than uncompressed formats. The current version uses a codec of libquicktime to make MPEG4, but the codec actually used is subject of change.

If you use the \*.avi files to create videos for presentation, test your player/decoder on the target platform for compatibility! IMPORTANT: Do not use a videoformat with lossy compression like this, if you want to process the video later, e.g. for pupil detection!

MPEG4 encoded AVI-files are probably your desired output for the head mounted cameras. However, the encoding puts a lot of load to the processor, so watch the counter of lost frames (tab "All Cameras").

# **G** Input Files

## G.1 Stimulation Profiles

EyeSeeCam is not only a video-oculography device, it can also act as a central control unit for physiological experiments in the course of which not only eye movements are measured, but the subjects are also stimulated by visual stimuli presented on a video beamer or by vestibular motion or galvanic stimuli that are generated by data aquisition hardware (USB-DUX). Other signals like the a posturography platform or a head tracker can also be measured synchronously to the eye movements.

The values on the analog outputs of the data aquisition hardware and the movement or position profiles of the visual stimuli have to be defined as a time series in a Matlab version 4 data file. Such data must adhere to a specific "syntax" which is described in this chapter.

The EyeSeeCam user can either use the predefined stimulation profiles or he can generate his own files in conjunction with a new experiment. Similar to experiments, the administration of stimulation profiles is also done in the Database (see Chapter 9 for details). If the file name of a stimuluation profile is inserted as an item to the "Profile" column of the Database table "Measurements" the measurement is started together with the thus defined stimulation sequence as soon as the "Start" button on the "Eyetracker / Record" tab is pressed.

#### G.1.1 Basic Contents and Syntax

A stimulation profile can be generated either in Matlab or in another program that can handle Matlab files (like octave). It is assumed that the user is comfortable with the Matlab syntax. At a minimum, such a file (e.g. profile.mat) must contain the following Matlab matrixes or variables<sup>1</sup>:

Algorithm 1 Matlab session for viewing the contents of the Matlab file profile.mat								
>> load profile.mat								
>> whos								
Name	Size	Bytes	Class	Attributes				
Comment	1x17	34	char					
Data	1000x3	24000	double					
DataArgs	3x13	78	char					
DataNames	3x6	36	char					
DataUnits	3x1	6	char					

<sup>1</sup>Possible Matlab variables are: Data, DataArgs, DataNames, DataUnits, VisualObjectX, Comment

The most important variable is  $Data^2$ ; it contains the time series and it can have a content similar to this one:

Algorithm 2 Matlab session for viewing part of the contents of the Matlab variable Data

>> Data Data = 0 0 0 0.0050 0.0314 0.0346 0.0100 0.0628 0.0691 0.0150 0.0941 0.1035 0.0200 0.1253 0.1378 . . .

The variable DataNames<sup>3</sup> contains as many names as strings in its rows as there are columns in Data. The names are used by EyeSeeCam to establish the proper mapping between a column in Data and the corresponding device. The contents of DataNames might look like this:

Algorithm 3 Matlab session for viewing the contents of the Matlab variable DataNames

DataNames = time usbdux usbdux

>> DataNames

The variable DataUnits<sup>4</sup> is similar to the variable DataNames except that it defines the units in which the values in the columns of Data are given. The contents of DataUnits might look like this:

Algorithm 4 Matlab session for viewing the contents of the Matlab variable DataUnits >> DataUnits DataUnits = s

v

The variable DataArgs<sup>5</sup> is again similar to the variable DataNames except that it defines ... The contents of DataArgs might look like this:

 $<sup>^{2}</sup>$ The number of Data columns is variable and it depends on the devices used.

<sup>&</sup>lt;sup>3</sup>Possible row values as names for DataNames are: time, usbdux, visual

<sup>&</sup>lt;sup>4</sup>Possible row values for DataUnits are: s (seconds), V (Volts), W (visual world coordinates)

<sup>&</sup>lt;sup>5</sup>Possible row values for DataArgs are: D/A Channel 1, D/A Channel 3

Algorithm 5 Matlab session for viewing the contents of the Matlab variable DataArgs

>> DataArgs DataArgs =

```
D/A Channel 1
D/A Channel 3
```

The main task of a profile designer will be to write appropriate Matalb scripts that generate a Matlab<sup>6</sup> file with the required matrices. A script that generates the mentioned variables might look like this:

Algorithm 6 Matlab script for generating a simple stimulation profile

```
Generate sinusoidal profiles for analog outputs of the USB-DUX
응
    clear all;
응응
   Sampling period (s)
    dt = 0.005;
%% Total duration of profile (s)
   T = 5;
%% Amplitude (V)
   A1 = 1; A2 = 1; A_text = '1.0';
    Comment = 'Amplitude = 1.0 V';
%% Create the time series and corresponding data
    t = (0:dt:(T - dt))';
    x1 = A1 * sin( 1*2*pi*t); % DUX-BOX DA channel 1
    x2 = A2 * sin(1.1*2*pi*t); % DUX-BOX DA channel 3
   Data = [t x1 x2];
%% Create Table Header
    DataNames = [
        'time ';
        'usbdux';
        'usbdux';
        ];
    DataUnits = [
        ′s′;
        'V';
        'V';
        ];
    DataArgs = [
                       ';
        'D/A Channel 1';
        'D/A Channel 3';
        ];
    % Note that only 2 channels are allowed,
    % DUX-BOX channelss 0 and 2 do not work properly
%% Plot and Save to File
   plot(Data(:,1),Data(:,2:end),'k-');
    xlabel([DataNames(1,:) ' [' DataUnits(1,:) ']']);
ylabel([DataNames(2,:) ' [' DataUnits(2,:) ']']);
    legend(DataArgs(2:end,:));
    save('profile.mat', '-v4', ...
     'DataNames', 'DataUnits', 'DataArgs', 'Data', 'Comment');
```

<sup>6</sup>It is mandatory to generate files that are compatible with version 4 of Matlab. This file format is simple, it was documented right from the beginning and other programs like octave can also handle it. Such a file can be generated by using the option '-v4' of the Matlab save command.



The results can be displayed with the Matlab command plot as in this figure:

Figure G.1: Visualization of the contents of a profile. In this example the two functional D/A channels of the DUX-BOX hardware will generate two sinusoidal voltage profiles of different frequencies and peak amplitudes of 1 V.

#### G.1.2 Laser Pattern Activation

One of the simplest functionalities that can be implemented in a stimulation profile is a time series containing activations / deactivations of the laser calibration pattern. This can be achieved with the following contents of a stimulation profile:

Algorithm 7 Matlab session for viewing the contents of a profile that switches the calibration laser on and off.

```
>> DataNames, DataUnits
DataNames =
time
usbdux
usbdux
DataUnits =
s
V
V
```

## G.1.3 Recording Voltage Signals from other Devices

Text

#### Posturography platforms

Text

#### G.1.4 Output of Voltage Signals for other Devices

Text

#### **Galvanic Vestibular Stimulation**

Galvanic Vestibular Stimulation (GVS) is a method that stimulates the labyrinthine (vestibular) nerve in the inner ear directly with currents on the order of a few mA. Normally, a voltage controlled bipolar current generator<sup>7</sup> [13] is used to apply the stimulation current transcutaneously by means of two electrodes on the mastoid processes. Such a generator can be driven by a voltage from an attached DUX-BOX. This way, the stimulation current can be synchronously recorded together with the eye movements, and, depending on other attached devices, additional signals like those from a posturographic platform can also be aquired. The Matlab script that was used to generate the results of Figure G.1 on the facing page is well able to create an appropriate GVS profile.

#### G.1.5 Recording an External Head Tracker

Text

#### G.1.6 Visual Stimulation

When another video monitor or beamer is connected to the external display connector of the EyeSeeCam laptop, EyeSeeCam can act as a visual stimulator. With an appropriate stimulus profile a full screen with a dark background is opened on the external video monitor and configurable visual motion patterns are displayed. Both the appearence of visual objects and the movement parameters of those objects can be configured in the profile. This way, eye movement data can be recorded synchronous to the visual stimulus and parameters like saccade metrics can be determined from the same data set.

First, the visual world needs to be set up in the variable VisualWorld. Parameters like screen size, type of projection, and distance of the subject from the screen are part of the visual world. A Matlab script snippet that generates an appropriate variable might look like this<sup>8</sup>:

<sup>&</sup>lt;sup>7</sup>The current stimulator proposed by [13] needs some minor modifications in order to work properly.

<sup>&</sup>lt;sup>8</sup>Those skilled in the art of OpenGL programming will immediately recognize the structure of the functions glFrustum and glOrtho in the way the parameters are defined. For further details on the meaning of these parameters a profile designer is referred to the documentation of these OpenGL functions.

8	Generate VisualWorld	d	
	VisualWorld = [		
	2.0;	8	Screen width in m (meter)
	1.5;	8	Screen height in m
	2.0;	00	Distance of subject from screen in m
	1.0;	8	Type of projection (1=perspective, 2=orthographic)
	-1.0;	8	left world coordinate of the near plane
	+1.0;	00	right world coordinate of the near plane
	75; s	8	bottom world coordinate of the near plane
	+.75;	00	top world coordinate of the near plane
	+2.0;	8	distance of viewpoint from the near plane in the world
	+5.0;	8	distance of viewpoint from the far plane in the world
	];		

Algorithm 8 Matlab script snippet for generating the variable VisualWorld

## G.2 Offline Analysis of Video Files

The PGM and RAW video files that have been created with EyeSeeCam (see Sections F.2 and F.3, respectively) can also be opened and analyzed offline by dragging and dropping a file to the EyeSeeCam Desktop icon. As soon as EyeSeeCam is started with this method the opened video file acts as a "virtual" Camera and EyeSeeCam becomes a video player.

## **Bibliography**

- D. Zhu, S. T. Moore, and T. Raphan, "Robust pupil center detection using a curvature algorithm," *Comput Methods Programs Biomed*, vol. 59, pp. 145–157, 1999.
- [2] A. W. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least squares fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, 1999.
- [3] E. Schneider, K. Bartl, S. Bardins, T. Dera, G. Böning, and T. Brandt, "Eye movement driven head-mounted camera: It looks where the eyes look," in *Proceedings of the IEEE Conference on Systems, Man and Cybernetics (SMC2005)*, Hawaii, USA, October 2005.
- [4] E. Schneider, K. Bartl, T. Dera, G. Böning, and T. Brandt, "Gaze-aligned head-mounted camera with pan, tilt and roll motion control for medical documentation and teaching applications," in *Proceedings of the IEEE Conference on Systems, Man and Cybernetics* (SMC2006), Taipei, Taiwan, 2006.
- [5] D. Tweed and T. Vilis, "Implications of rotational kinematics for the oculomotor system in three dimensions," *J Neurophysiol.*, vol. 58, no. 4, pp. 832–849, Oct 1987.
- [6] W. Haustein, "Considerations on listing's law and the primary position by means of a matrix description of eye position control," *Biol Cybern.*, vol. 60, no. 6, pp. 411–420, 1989.
- [7] J. B. Pelz and R. Canosa, "Oculomotor behavior and perceptual strategies in complex tasks." *Vision Res.*, vol. 41, no. 25-26, pp. 3587–3596, 2001.
- [8] H. T., "Mathematics of three-dimensional eye rotations," *Vision Res.*, vol. 35, no. 12, pp. 1727–1739, Jun 1995.
- [9] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*. Berkley: University of California Press, 1967, pp. 281–297.
- [10] J. Kim, "A simple pupil-independent method for recording eye movements in rodents using video," *J Neurosci Methods*, 2004.
- [11] E. Schneider, S. Glasauer, and M. Dieterich, "Comparison of human ocular torsion patterns during natural and galvanic vestibular stimulation," *J Neurophysiol*, vol. 87, pp. 2064– 2073, 2002.

- [12] D. Ott, F. Gehle, and R. Eckmiller, "Video-oculographic measurement of 3-dimensional eye rotations," *J. Neurosci. Methods*, vol. 35, no. 3, pp. 229–234, 1990.
- [13] W. S. Woodward, "Optically isolated precision bipolar current source," *Electronic Design*, vol. 46, p. 130, 1998. [Online]. Available: http://www.elecdesign.com/Articles/Index. cfm?ArticleID=6304

# Index

3D, 12, 57 500Hz, 12 Acrobat Reader, 37 adjustment, 10, 11 alone, 33 analog outputs, 46, 77 Apple, 9 AVI-File, 76 battery, 9 beamer, 46, 77, 81 binocular, 11 blink, 58 browse, 33 cable, 10 calibration, 12, 80 column, 78 configuration, 71 Conjuncain, 57 connector, 9 contact lens, 57 cosmetic, 12, 57 current generator, 81 Data, 78 data aquisition, 77 data browsing, 35 DataArgs, 78 Database, 77 DataNames, 78 DataUnits, 78 desktop, 15 duration, 46 DV-File, 76 elastic, 11

electrodes, 81 eps, 33, 36 eval, 38 EvalEZPlot, 38 experiments, 77 export, 36 exported, 33 external display, 81 external viewer, 34 eye drops, 57 ezplot, 33 figure, 39 file, 33 file browser, 33 file name, 37 focal length, 13 focusable, 12 fps, 71 frames per second, 71 full screen, 81 galvanic, 46, 77 Galvanic Vestibular Stimulation, 81 goggles, 10 GUI, 15 GVS, 81 Hand, 34 head position, 58 head tracker, 77 height, 71 horizontal, 34 idle, 70 IEEE1394, 9 Illustrator, 36

image file, 36 injection, 57 inner ear, 81 invasive, 57 iris, 57 kpdf, **37** landmarks, 12, 57 Laptop, 9 laptop, 81 laser, 12, 80 LeftEyeHor, 28 LeftEyeTor, 28 LeftEyeVer, 28 LeftPupilCol, 29 LeftPupilEllipseCol, 29 LeftPupilEllipseRow, 29 LeftPupilRow, 29 LeftTime, 28 lenses, 12 limbus, 58 line of sight, 12 Linux, 9 marker, 17, 71 mastoid, 81 Matlab, 33, 46, 77 Matlab file, 33 Matlab matrix, 77 Matlab syntax, 77 Matlab variable, 77 measurement, 77 Measurements, 46 monucular, 10 needle, 57 nerve, 81 optokinetic, 46 original view, 36 password, 69 pdf, 33, 36

PGM-File, 73, 82

pigment, 12, 57 Plot, 33 posturography, 77, 81 power supply, 9 PowerPoint, 36 print, 36 printed, 33 printer, 36 Profile, 46 Profiles, 77 prompt, 39 raw, 38 RAW-File, 75 rows, 78 rubber, 12 s. 11 saline solution, 57 scientific data plots, 36 sclera, 12, 57 script, 79 scroll, 35 sequence, 46 shell, 34 silicone, 12 sklera, 17 speed, 71 spring, 12 stand-alone, 39 sterile, 12, 57 stimulation profile, 47 stimuli, 77 strings, 78 subplot, 34 suffix, 37 surgical pen, 12, 57 synchronized, 36, 46 synchronously, 77 syntax, 46 System, 47 system database, 71 Time, 28 time, 78

time axes, 35 time series, 77, 78 toolbar, 35 torsional, 34 transcutaneous, 81 uncalibrated, 38 USB-DUX, 46, 77 usbdux, 78 velocities, 38 vertical, 34 vertical range, 12 video monitor, 81 videobox, 15 viewer, 34 visual, 78 visual stimulator, 81 visual stimulus, 46, 77 voltage, 81 wash out, 58 Word, 36xterm, 39

Zoom, 33 zoom, 35